

Synthesis of Separation Systems for Multicomponent Product Problems

Eoin Cathal McCarthy

Thesis presented for the Degree of Doctor of Philosophy

University of Edinburgh

2000



Acknowledgements

This submission of the thesis has been a long time coming. I would like to the following for helping to get me there:

- My PhD supervisors, Professor Jack Ponton and Dr Eric Fraga for giving me the opportunity to do this work, for their patience and for their support throughout.
- The Engineering and Physical Sciences Research Council for the funding for this project.
- My parents and brothers for their love and encouragement without which I would not have seen this through.
- Ross and Greville, my officemates of the last 4 years, for putting up with my (occasional) ranting at the computer.
- My friends Colm Fitzgerald, Adam Harvey, Christine Merrick and many others in Edinburgh and in Ireland for their encouragement and for reminding me that there is much more to life than a PhD.

Abstract

Separation systems are of great importance to the operation and profitability of chemical processes. They are responsible for ensuring that the process meets environmental and product specifications. Computer-aided synthesis methods play a large role in the development of process plants and in the reduction of product time-to-market. Such methods have been successfully developed for separation system problems. However, their focus has largely been on pure component products. There exist separation processes, particularly in the petroleum industry, which are concerned with the manufacture of multicomponent products (MCPs). Many of these are *effect* products, i.e. products whose specifications are in the form of a property such as density or vapour pressure and therefore are satisfied by a region of compositions. As a result the search for good solutions to MCP separation problems requires a much greater exploration of stream space than is undertaken for pure component product problems. This in turn leads to opportunities for the use of non-sharp separation, stream splitting and blending.

This thesis presents a method for the synthesis of multicomponent product separation systems which does not require the designer to provide an explicit superstructure. Separator functions are not predetermined and systematic stream splitting and blending are included. The synthesis algorithm employs a depth-first tree search in order to locate solutions, and unit design variable discretisation to reduce the search space. The algorithm generates a set of good, feasible solutions which may be further optimised by continuous means.

The method is applied to two different types of problems. The first of these involves ideal-, near-ideal multicomponent product separations. Two such problems are drawn from a literature source and demonstrate that the method can generate new and known solutions to these problems. The final MCP separation problem presented involves the separation of a two-phase oil stream into specified gas and liquid products. An effect product specification, in the form of vapour pressure bounds, is placed on the liquid product. Flash vessels are the separation technology considered and all solutions returned make use of stream blending. This is a separation problem of industrial scale and interest which most literature approaches would be unable to attempt.

Nonideal separations are the second problem type considered. Due to the limits of separation imposed by the presence of azeotropes and the possibility that the recycles required to make these processes optimal can be multicomponent streams, these problems also require a greater exploration of stream space in the search for good solutions. The MCP synthesis method is applied to nonideal problems by adding a procedure for targetting feasible separations, a shortcut column design and the means to handle recycles. Effect product specifications are defined for the recycle streams and solutions containing entrainer-rich and azeotrope-rich recycles and boundary-crossing separations are returned for ternary nonideal separation problems.

Contents

1	Introduction	1
1.1	Process Synthesis	1
1.2	Multicomponent Product Separation Synthesis	3
1.3	Extending the MCP Synthesis Method to Nonideal Separation Problems	6
2	MCP Synthesis Literature Review	8
2.1	Introduction	8
2.2	Definition of MCP process operations	8
2.2.1	Separation definitions	9
2.2.2	Splitting and blending definitions	10
2.3	The Representation Problem	11
2.3.1	Material Allocation Diagram	12
2.3.2	Component Recovery Matrix	14
2.3.3	Component Assignment Diagram	16

2.3.4	Stream Allocation Diagram	18
2.3.5	Separation Product Matrix	21
2.3.6	Superstructure	24
2.3.7	Summary and Conclusions	25
2.4	Evaluation Problem	25
2.5	Synthesis Strategy	27
2.5.1	Heuristic Strategies	27
2.5.2	Evolutionary Strategies	28
2.5.3	Algorithmic Strategies	29
2.5.4	Synthesis strategies - summary and conclusions	30
2.6	Conclusions	31
3	MCP Synthesis Method	34
3.1	Introduction	34
3.2	State definitions	34
3.3	Structural alternative generation	36
3.3.1	The framework for structure generation	37
3.3.2	Rules for the addition of process units	39
3.3.3	Defining structure generation operations	43

3.3.4	Structure generation algorithm	48
3.4	Unit design space	51
3.4.1	Unit design alternatives	52
3.4.2	Providing the unit design information	53
3.5	Searching for solutions	53
3.5.1	The synthesis tree	54
3.5.2	Selection of search method	55
3.5.3	Search algorithm	56
3.6	Removal of redundancies	59
3.7	Completeness of the structure generation method	63
3.7.1	CT unit structural alternatives	63
3.7.2	Blending structural alternatives	65
3.8	Summary and conclusions	68
4	MCP Synthesis Results	69
4.1	Introduction	69
4.2	Unit models	69
4.2.1	Thermodynamic and physical properties	70
4.2.2	Flash model	70

4.2.3	Compressor units	72
4.2.4	Distillation column model	73
4.3	Literature problems	74
4.3.1	Problem 1	75
4.3.2	Problem 2	80
4.4	Oil stabilisation problem	87
4.4.1	Problem input	87
4.4.2	Discussion of results	96
5	Nonideal Separation Background	98
5.1	Introduction	98
5.2	Nonideal mixtures and azeotrope formation	98
5.3	Representation of Nonideal Systems	99
5.4	Feasible separations in nonideal ternary systems	102
5.5	Entrainers	103
5.6	Nonideal Separation Synthesis Methods	105
5.7	Contributions to be made	108
6	Nonideal Separation Synthesis	109

6.1	Introduction	109
6.2	Targetting nonideal separations	109
6.2.1	Modelling distillation line maps	110
6.2.2	The desired separation alternatives	111
6.2.3	Setting the design specifications for the targetted separations . .	112
6.3	Modelling Azeotropic Separations	118
6.3.1	Specifications for the model	119
6.4	Introduction of Recycles	122
6.4.1	Entrainer recycles	123
6.4.2	Azeotrope recycles	126
6.4.3	Enumerating the recycle alternatives	134
6.5	Summary and conclusions	139
7	Nonideal Separation Synthesis Results	140
7.1	Introduction	140
7.2	Acetone chloroform separation	140
7.2.1	Feed on concave side of distillation boundary	143
7.2.2	Feed on convex side of distillation boundary	146
7.3	Toluene Isopropyl-alcohol separation	155

7.3.1	Feed on concave side of distillation boundary	156
7.3.2	Feed on convex side of distillation boundary	160
7.4	Discussion of results	162
8	Conclusions	164
8.1	Introduction	164
8.2	Multicomponent product synthesis	165
8.3	Nonideal separation synthesis	168
8.4	Recommendations and Future Work	170
8.4.1	MCP Synthesis	170
8.4.2	Nonideal separation synthesis	171
A	Implementation of the MCP Synthesis Method	172
A.1	Introduction	172
A.2	Structuring the synthesis problem information	172
A.2.1	Synthesis problem hierarchy	173
A.2.2	Unit models	178
A.2.3	Incorporating unit models into synthesis	179
A.3	Search objects	179

A.4	Structure generation operations	181
A.5	Operator and solution handling	182
A.6	The search algorithm	184
B	Targetting Nonideal Separations	186
B.1	Introduction	186
B.2	Modelling distillation line maps	186
B.3	Integrating targetted alternatives into the synthesis problem state . . .	191
C	Nonideal Separation Column Model	193
C.1	Introduction	193
C.2	Determination of the minimum number of stages	193
C.2.1	Setting up	194
C.2.2	Minimum number of stages algorithm	195
C.2.3	Mass balance algorithm	195
D	Acroynms	197
	References	199

List of Figures

1.1	Structure of thesis	7
2.1	Structure of multicomponent product flowsheet	11
2.2	Stream splitting	11
2.3	Stream bypassing	11
2.4	Stream blending	11
2.5	Example Material Allocation Diagram	13
2.6	Sharp separation on MAD	13
2.7	Splitting along MAD profile	13
2.8	Stream blending on MAD	13
2.9	Example component assignment diagram	17
2.10	Decision tree for CAD example	17
2.11	Example stream allocation diagram	19
3.1	Splitting the feed stream	41

3.2	Adding the separator	41
3.3	Splitting the lone intensive stream states	42
3.4	Taking the blending opportunity	43
3.5	Internal blending structure	47
3.6	The cycle of operations	49
3.7	Adding more information to the problem state	51
3.8	Discretisation of a unit variable across its range	52
3.9	Synthesis tree	54
3.10	Depth-first tree search algorithm	57
3.11	Internal blending redundancies	60
3.12	CT redundancy - first type	62
3.13	CT redundancy - second type	62
3.14	An direct separation sequence	64
3.15	Creating a parallel structure	65
3.16	Block diagram of separation process	65
3.17	Blending of four feed streams	66
3.18	Product bypass alters product specifications	68
4.1	Aggarwal and Floudas' best solution for problem 1	77

4.2	Semi-sharp solution to problem 1	78
4.3	Best nonsharp solution obtained for problem 1	79
4.4	Aggarwal and Floudas' best solution for problem 2	81
4.5	Best nonsharp separation solution for problem 2	83
4.6	Third best nonsharp separation solution for problem 2	84
4.7	Best solution with single distillation column	86
4.8	Second best solution with distillation column	87
4.9	Oil stabilisation solution 1	91
4.10	Oil stabilisation solution 2	91
4.11	Oil stabilisation solution 3	92
4.12	Oil stabilisation solution 5	92
4.13	Oil stabilisation solution 12	93
4.14	Best compressor design for solution 1	94
4.15	Best compressor design for solution 12	95
5.1	Acetone-IPA-Toluene distillation line map	100
5.2	Stable fixed point	101
5.3	Unstable fixed point	101
5.4	Saddle fixed point	101

5.5	Separation region of a ternary ideal mixture	103
5.6	Separation region of a ternary nonideal mixture - no distillation boundary	104
5.7	Separation region of a ternary nonideal mixture - with distillation boundary	104
5.8	Boundary-crossing separation regions	105
6.1	Nonideal system information hierarchy	111
6.2	Nonideal separation alternatives	113
6.3	Checking if the boundary is concave wrt the feed	115
6.4	Testing if the separation feed is inside the boundary-crossing region . . .	116
6.5	Determining the maximum extent of an equilibrium conode separation	118
6.6	Direction of stagewise calculations	121
6.7	Example where total reflux operation is infeasible	122
6.8	Influence of magnitude of entrainer flowrate on separation processes . .	124
6.9	The mixing goal preserves the separation task	126
6.10	Binary azeotropic separation taken to its limits	127
6.11	The limit of separation and the azeotrope recycle	129
6.12	Limit of separation for entrainer-feed mixing	131
6.13	Limit of separation for entrainer-azeotrope mixing	132

6.14	Limit of separation is the binary azeotrope	132
6.15	Adjusting entrainer flowrate to preserve the limit of separation	133
6.16	Including recycle operations in cycle of structure generation operations .	138
6.17	Structure created by recycle operations	138
7.1	Acetone, chloroform, benzene residue curve map	141
7.2	Concave feed acetone-chloroform separation - Solution 1	148
7.3	Concave feed acetone-chloroform separation - Solution 3	149
7.4	Concave feed acetone-chloroform separation - Solution 7	150
7.5	Concave feed acetone-chloroform separation - Solution 8	151
7.6	Concave feed acetone-chloroform separation - Solution 9	152
7.7	Convex feed acetone-chloroform separation - Solution 1	153
7.8	Convex feed acetone-chloroform separation - Solution 18	154
7.9	Residue curve map for acetone,IPA,toluene system	155
7.10	Concave feed IPA-toluene separation - Solution 1	157
7.11	Acetone-IPA-Toluene concave feed separation - new solution structure .	159
7.12	Acetone-toluene-IPA separation: convex feed-specific solution 1	161
7.13	Acetone-toluene-IPA separation: convex feed-specific solution 2	162

A.1	Synthesis problem information hierarchy	174
A.2	Algorithm for performing unit design	180
A.3	Organisation of structure generation operations	182
A.4	Organisation of operator and solution handling	183
B.1	Nonideal system information hierarchy	187
B.2	Position of targetting in the synthesis hierarchy	192
C.1	Infinite reflux algorithm	194

List of Tables

2.1	Example R-matrix	15
2.2	Sharp separation on example R-matrix yields two new R-matrices	15
2.3	Representation features	25
2.4	Synthesis strategy features	31
4.1	Aggarwal and Floudas Problem 1 - Feed stream component flows	75
4.2	Product set data	76
4.3	Product 1 specifications	76
4.4	Product 2 specifications	76
4.5	Nonsharp solution costs for problem 1	79
4.6	Aggarwal and Floudas Problem 2 - Feed stream component flows	80
4.7	Product set data	81
4.8	Product 1 specifications	81
4.9	Product 2 specifications	81

4.10	Column designs for Aggarwal and Floudas' solution	82
4.11	Costs of nonsharp separation solutions to problem 2	84
4.12	Costs of semi-sharp, non-sharp solution set to Problem 2	85
4.13	Feed stream data	88
4.14	Liquid product specifications	88
4.15	Gas product specifications	89
4.16	Flash design variable data	89
4.17	Compressor design variable data	89
4.18	Costs of oil stabilisation solutions	90
4.19	Improved compression oil stabilisation solutions	95
7.1	Acetone product specifications	142
7.2	Chloroform product specifications	142
7.3	Wilson parameters for acetone-chloroform-benzene system	142
7.4	Acetone-chloroform separation: concave feed data	143
7.5	Costs of solutions to concave binary feed separation	144
7.6	Acetone-chloroform separation: convex feed stream data	146
7.7	Costs of solutions to convex binary feed separation	146
7.8	Wilson parameters for Acetone-IPA-toluene system	155

7.9 IPA-toluene separation: concave feed stream data 156

7.10 Costs of solutions to concave binary feed separation 156

7.11 Acetone-Toluene-IPA concave feed separation - new solution structure
column designs 158

7.12 Feed stream data 160

Chapter 1

Introduction

1.1 Process Synthesis

Nishida et al. [30] have defined process synthesis as “the act of determining the optimal interconnection of processing units as well as the optimal type and design of the units within the processing system”. According to the same authors three important problems exist in process synthesis - representation, evaluation and strategy.

Representation

The representation of a synthesis problem consists of the description of the problem and the operations defined on that description. The set of operations available give rise to the process alternatives. The aim is to develop a representation which is rich enough to allow all alternatives to be included and ”clever” enough to exclude automatically ridiculous options. It is also desirable if the representation can aid directly in solving the synthesis problem.

Evaluation

A chemical process is a very complex system. Therefore, it is not realistic to attempt rigorous solutions to a process synthesis problem from the very start. Design usually proceeds in stages, which increase in rigour and detail but deal with fewer process

alternatives. In the early design stage, the aim is to generate a set of feasible (hopefully good) solutions coupled with a reduction of the search space through the elimination of infeasible alternatives and the worst solutions. Given the initial large size of the search space and the emphasis on feasibility rather than rigour, this calls for the use of fast, approximate evaluation of each process alternative, eg. shortcut costings and unit models. In contrast, the detailed design stage requires rigour and the unit designs and costings must reflect this. The effort expended on the evaluation of alternatives is tied to the desired accuracy of the solutions which is in turn dependent on the design stage.

Strategy

The goal of synthesis strategy is to locate quickly the better alternatives without totally enumerating all options. It is clear from this that synthesis is a search problem. Various methods which have been used either independently or in combination in process synthesis are discussed by Sheppard [35]. These methods may be applied within a heuristic, evolutionary or algorithmic framework.

Heuristics are rules drawn from experience and understanding of a certain problem and can be used to either generate or select process alternatives. Because of their basis in experience, heuristics tend to value some structures, units or unit designs over others, prior to the evaluation of the objective function. Therefore, they are best used on the types of problems for which they were originally formulated.

The idea of *evolutionary design* has been central to the modus operandi of process engineers for the improvement of chemical process systems. It consists of making a sequence of design modifications to a previously synthesized process, leading to an improved design. Since processes synthesized by evolutionary methods depend heavily on the initial flowsheets, it is very important for these initial flowsheets to be "reasonable" in order to eventually lead to optimal or near-optimal flowsheets. Evolutionary rules are used to generate all the permissible changes to a flowsheet. Evolutionary design is needed because our knowledge is incomplete. It cannot be guaranteed that the design found satisfies the design constraints and is optimum. However, the more knowledge

utilised, the less evolution or modification that should be necessary, i.e. the better a designer understands a problem, the better he/she can choose modifications which will improve the process design.

An algorithm is a mathematical problem formulation and solution method. Whereas heuristics may implicitly value one structure over another before an objective function is evaluated, a rigorous algorithmic method does not. As a result, algorithmic approaches explore larger sections of the solution space for a given problem, placing greater demands on computing power and time. However, this brings the potential for a better solution compared to heuristic or evolutionary methods. If a knowledge base for a problem is lacking then an algorithmic method may yield better results. Conversely, it is possible to incorporate knowledge into the algorithm so as to reduce the solution space and generate better designs more quickly (Raman and Grossman, [32]). Methods which make use of a nonlinear programming approach (Floudas, [11], Smith and Pantelides, [37]) may require a feasible starting point, i.e. a feasible process, in order for the nonlinear solver to perform well. Implicit enumeration approaches such as CHiPS (Fraga and McKinnon, [13]) need only the problem formulation to attempt a solution.

1.2 Multicomponent Product Separation Synthesis

Separation systems are of key importance to the operation and profitability of chemical processes. They are largely responsible for ensuring that the process meets environmental and product specifications. There has been much work in the area of separation system synthesis aimed at developing methods and tools to reduce the design time and the workload on the designer. The focus of the work has been almost exclusively on pure component product problems.

There exist separation processes, particularly in the petroleum industry, which are concerned with the manufacture of multicomponent products. These products are not usually specified simply by composition. Rather, the specification takes the form of a

desired *effect* or property such as vapour pressure, viscosity or density. Such specifications may be satisfied by a region in stream composition space. Effect products are also found in the waste treatment industry where specifications such as biochemical oxygen demand (BOD) and chemical oxygen demand (COD) quantify the effect of a stream on the environment. Wahnshafft et al. [44] formulate effect product specifications for multicomponent recycle streams occurring in separation processes.

For a pure component product separation it is known that good solutions occupy a small portion of the available stream composition space, namely those regions close to the pure components. As a result, synthesis methods tend to explore only those areas. The composition regions which provide good solutions to multicomponent product problems are not known, not least because the possible product compositions are unlimited. Therefore, a more thorough exploration of stream space is required to determine the best solutions. The requirements of a synthesis method to achieve this are described below:

- *Wider unit operating ranges*

Multicomponent products can of course be produced by separating the feed into its constituent components and blending subsequently. Such a process does more separation work than is necessary and the act of mixing streams is thermodynamically unsound as entropy is being created. To avoid this, a wider operating range for separators must be considered, i.e. separations with component recoveries lower than 85-100% in the output streams must be made available. This opens up the possibility of using single-stage equilibrium separators such as flash vessels.

- *Systematic stream splitting and blending*

Splitters and blenders can be used to access compositions between those of the feed stream set and the separation outputs. Blending may be unavoidable where the “effect” property differs from the property upon which the separation is based. Stream splitting will generate structures with parallel paths. Such structures may be useful provided that they do not contain many small separation units, as this

will tend to violate economies of scale.

- *Alternative generation and evaluation*

Experience of processes involving low recovery distillation separations, stream splitting and blending is very limited. Heuristics and evolutionary rules rely on such experience. Hence synthesis methods based primarily on heuristic or evolutionary rules are unlikely to be useful. The generation and evaluation of process alternatives will require to be deterministic and systematic.

Jaksland *et al* [18] present a synthesis method for separation processes based on the analysis of physicochemical properties and their relationships to separation techniques for the construction of *physically feasible* process flowsheets. It is capable of considering single stage, low recovery separations eg. flashing, crystallization. The emphasis is on the consideration of the possible separation technologies and is different from the present work whose aim is the automated generation and exploration of structural and design alternatives for a given set of separation technologies.

The intention of the present work is to provide a method for the preliminary design of MCP separation processes. This is the stage where the most process alternatives exist and an effective early design method is one which removes much the workload from the designer, whose main concern is the selection of candidate processes for the next, more rigorous, stage of design. This is achieved through automation of the synthesis process, i.e. alternative (structural and design) generation, evaluation and searching. The ability to return a set of solutions is also highly desirable as a single solution is not likely to possess the best performance with regard to a range of considerations such as controllability, safety and environmental impact. This thesis presents a method for MCP synthesis which meets these requirements.

1.3 Extending the MCP Synthesis Method to Nonideal Separation Problems

Separation of nonideal mixtures into pure components is a problem commonly encountered in the chemical process industry. Much research has been conducted towards the understanding of nonideal separations. This research effort has yielded the thermodynamic models needed to describe nonideal behaviour and extensive data on the parameters required for their use. Building on these are the tools used for the analysis of nonideal VLE systems (residue curve maps [9], distillation line maps [40]) which have yielded knowledge of the feasible separations available for a given feed composition and pressure. Finally, much work has gone into determining the different types of entrainers which are used to facilitate the separation of azeotropes [1, 20]. This research makes the modelling of nonideal separations and the determination of process alternatives possible, setting the stage for the development of synthesis methods for nonideal separation processes.

Nonideal separation problems possess limits on separability, in addition to pure components, imposed by the presence of azeotropes. Recycles are required and the streams involved are generally multicomponent streams. Wahnschafft et al. [44] formulate the recycles in terms of an *effect* product. Again, as with MCP synthesis problems, the exploration of stream space is not limited to the regions close to the pure components. Extending the MCP synthesis method to encompass ternary azeotropic separation problems requires that the limits of separation of a given feed can be determined, that recycles can be incorporated, and that the separations can be modelled and costed. The work on MCP synthesis is built upon to provide the ability to solve ternary nonideal separation problems.

An overview of the thesis structure is given in figure 1.1.

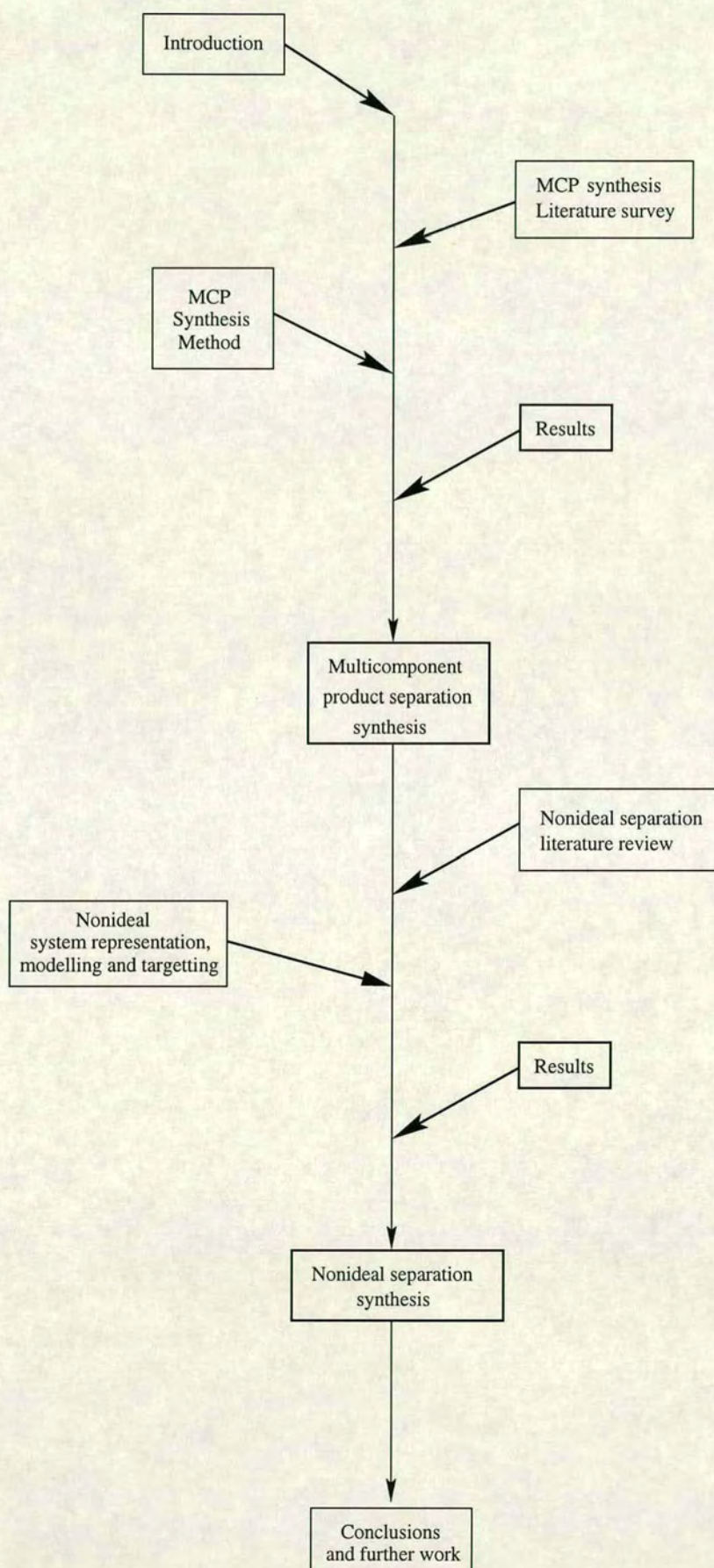


Figure 1.1: Structure of thesis

Chapter 2

Multicomponent Product Synthesis Literature Review

2.1 Introduction

This chapter reviews research by other authors on multicomponent product synthesis methods. It starts with the definitions of the terms used in the analysis of the literature approaches. Next, the literature methods are compared by their treatment of the three problems involved in process synthesis: representation, evaluation and strategy, as outlined in section 1.1. Finally, the features of each method are summarised and compared before the possible contributions to be made to multicomponent product synthesis are discussed.

2.2 Definition of MCP process operations

In order to facilitate the analysis of the literature methods, the operations which arise in multicomponent product processes will be defined.

2.2.1 Separation definitions

Distillation is the main separation technology considered for the literature MCP synthesis work and the assignment of *key components* is useful in characterising these separations. For this work the *light key* is defined to be the lightest component which may be present in the bottoms product in a significant amount. The *heavy key* is the heaviest component which may be present in the tops in a significant amount.

Sharp separation is the perfect separation between the light and heavy key components with each being recovered 100% in the tops and bottoms respectively. Non-keys lighter than the light key are totally recovered in the top product. Non-keys heavier than the heavy key are totally recovered in the bottom product.

In *semi-sharp separation* the key components may distribute between the top and bottom products. Non-keys lighter than the light key are assumed to be totally recovered in the top product and non-keys heavier than the heavy key totally recovered in the bottom product. The lower limit on the key recoveries such that the non-key conditions apply is usually in the region of 80% - 95%. This should be tested in individual cases.

For the case of *non-sharp separation* all components may distribute between the top and bottom products. Key components do not characterise the nonsharp separation as effectively as for the last two cases.

The *separation breakpoint* is a description of separator function. For a volatility-ordered list of components, the breakpoint is the separation between a pair of adjacent components. The lighter component of the pair corresponds to the light key and the heavier component to the heavy key. For example, a breakpoint of 1 describes the separation between the first and second components in the ordered list.

2.2.2 Splitting and blending definitions

Splitting and blending operations are best described in the context of the overall flowsheet structure. Figure 2.1 is a general representation of an MCP flowsheet and is composed of a set of feed streams which enter the separation section, within which splitting and blending of streams may also occur. A splitting and blending block can exist outside the separation section for the purpose of distributing the outputs of that section to obtain the final product set. Splitting and blending operations can be distinguished on the basis of whether they occur inside or outside the separation section.

Stream splitting refers to the division of a stream into two or more fractions of the same composition. Two cases apply:

1. Both fractions are available for further processing by separators (figure 2.2).
2. One output goes directly to the product blenders or direct to product. The other output fraction requires further separation. This operation is referred to as *stream bypassing* (figure 2.3). Bypassing has a particular trade-off: the greater the amount bypassed to a product, the sharper the separations required to obtain that product, but the lower the amount of material which must be separated. A maximum bypass fraction of a stream to a given product can be determined. At that fraction, a sharp separation task is created and the feed flow to the resultant separator is at a minimum.

Stream blending takes two or more input streams and mixes them to form a single output stream (figure 2.4). Two types of blending can be distinguished, the first of which is *internal blending* where the output from a blender goes to further separation. The output from *external blending* is a product or is destined for a product blender.

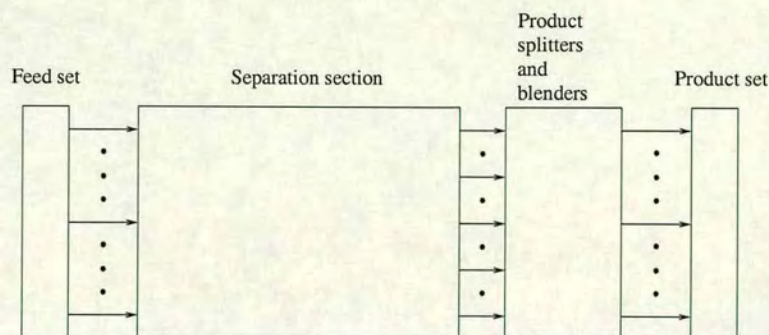


Figure 2.1: Structure of multicomponent product flowsheet

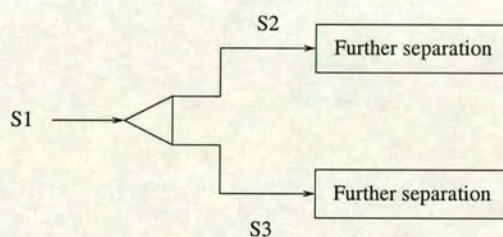


Figure 2.2: Stream splitting

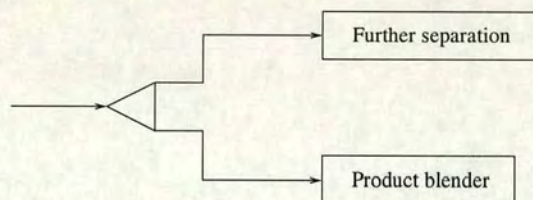


Figure 2.3: Stream bypassing

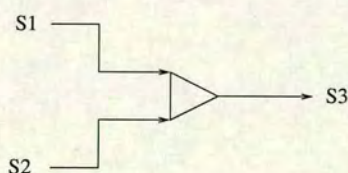


Figure 2.4: Stream blending

2.3 The Representation Problem

“Can a representation be developed rich enough to allow all alternatives to be included and clever enough to exclude automatically ridiculous options ? Can it also be clever enough to aid one directly in solving the synthesis problem ? ” (Nishida *et al.* [30])

The representation of a problem is of key importance to a synthesis method as this es-

establishes the search space for that problem. The representations used for the MCP synthesis problem fall into three categories : graphical, matrix-based and superstructure. Most variations of the first two types describe the distribution of the feed components in the product streams. While the information they contain may be the same, the approaches are distinguished by the manner in which that information is interpreted and used. In every case the components are ranked in order of the physical property upon which separation is based. It is assumed that this order does not change.

2.3.1 Material Allocation Diagram

The Representation

The Material Allocation Diagram (MAD) [27–29] is a graphical representation of a synthesis problem. It can deal with problems involving a single feed and two or more products. Components are represented by rectangles of fixed width and variable height. The height of a component for a given product is its recovery in that product. The MAD shows graphically the relation between the feed and the products. Two versions of the MAD are used by Muraki and coworkers for different multicomponent synthesis problems. The first is employed for the case of a single multicomponent feed, two multicomponent products (1R2P) process and the other for a single feed, greater than two products (1RMP) process. The MAD for the 1R2P process is unique. However, this is not the case for the 1RMP process. To overcome this, M 1R2P processes are created which separate the raw material into each individual product and the remaining products resulting in M MADs. A procedure involving product decomposition and stream matching is employed to generate a graphical relationship between the outflowing streams from a process flowsheet and the feed stream. The authors refer to this as the modified MAD or mMAD. Figure 2.5 illustrates the MAD for a 2 product, 5 component problem.

Operations defined on the representation

Stream splitting (figure 2.7), blending (figure 2.8) and separation can be represented

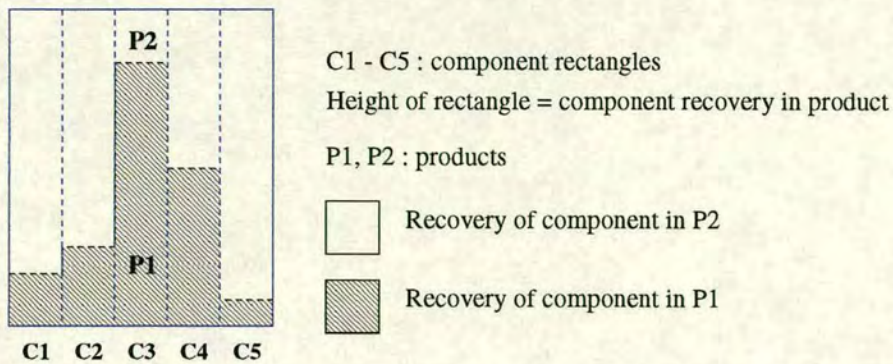


Figure 2.5: Example Material Allocation Diagram

using the MAD and mMAD. Semi-sharp and non-sharp separations can be shown but are difficult to manipulate. Sharp separations are easily captured by this representation (figure 2.6).

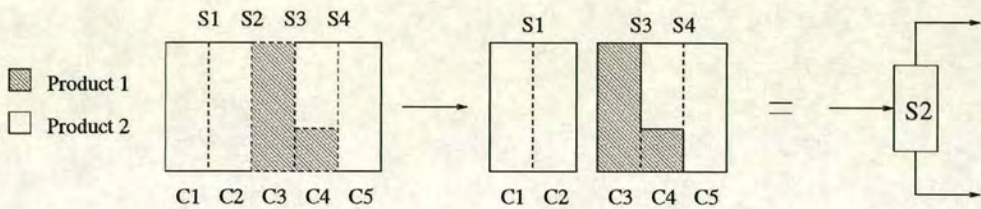


Figure 2.6: Sharp separation on MAD

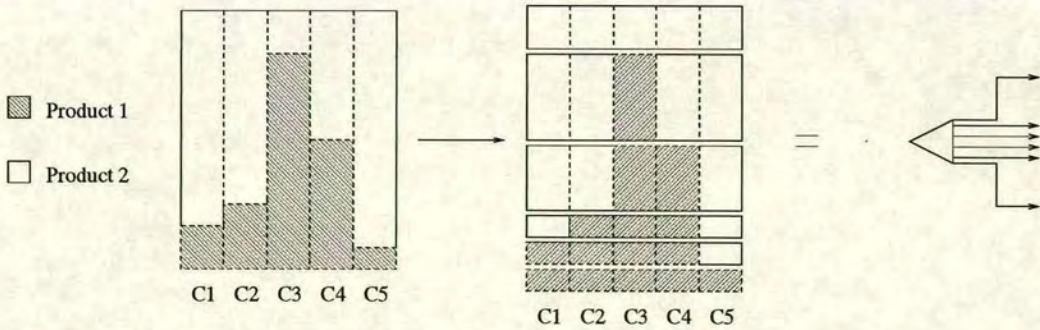


Figure 2.7: Splitting along MAD profile

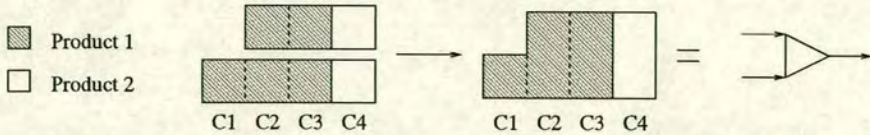


Figure 2.8: Stream blending on MAD

Synthesis information drawn from the representation

Muraki et al [27–29] obtain split fraction alternatives for synthesis by division of the MAD or mMAD along their profiles (see figure 2.7). During synthesis, a given sequence of separators is optimised by adjusting the amount separated at each separation breakpoint. The split fractions suggested by the MAD are searched to perform this optimisation. Manipulations of the MAD performed in accordance with the separator sequence and the chosen split fraction values to check to feasibility of each process.

The MAD and mMAD are most useful when only sharp separations are employed in the synthesis of processes. For this case separation problems and their manipulation take on a discrete nature and the diagrams are effective as a synthesis tool. The construction of the mMAD is dependent on knowing the composition of outlet streams from each separator. Therefore, the order and function (described by adjacent key components) of the separators must be set before synthesis, i.e. an initial flowsheet must be provided. The mMAD cannot be formed from a flowsheet containing nonsharp separators, as the outflowing streams may consist of all feed components. Neither the MAD nor the mMAD aid in suggesting alternative flowsheets. A problem may be represented using the diagrams but the diagrams do not give any clue as to the structures of alternative flowsheets available.

2.3.2 Component Recovery Matrix

The Representation

The Component Recovery Matrix or R-matrix, Bamopoulos *et al.* [4], is used to represent the product specifications. It can handle single feed, two-or-more product problems. Each element R_{ij} is the recovery of component i in product j . An example R-matrix is shown below.

Component	Product			
	1	2	3	4
A	0.6	0.4	0.0	0.0
B	0.5	0.5	0.0	0.0
C	0.2	0.0	0.8	0.0
D	0.0	0.0	0.4	0.6

Table 2.1: Example R-matrix

Component	Product		Component	Product		
	1	2		1 ₂	3	4
A	0.6	0.4	C	0.2	0.8	0.0
B	0.5	0.5	D	0.0	0.4	0.6

Table 2.2: Sharp separation on example R-matrix yields two new R-matrices

Operations defined on the representation

The operations of sharp separation, semi-sharp separation and feed stream bypassing are represented by manipulations of the R-matrix. For the example provided, a sharp split between components B and C results in the generation of the two new R-matrices shown in table 2.2. The operation of feed bypassing is defined on the R-matrix and the maximum amount that may be bypassed to a given product i , $f_{i,max}$ is equivalent to the value of the minimum recovery of a component in that product. After each operation the rows of each submatrix are normalised to 1.0. Thus each matrix for a given problem tells us how much of each component in that matrix must be recovered in each product. However, this means that stream blending and splitting (other than bypassing) cannot be represented without ambiguity. The problem description is not unique for these operations as the R-matrix is only interested in the relationship between the subproblems and the products. The normalisation operation removes the information on the subproblem flows needed for the splitting and blending operations.

Synthesis information drawn from the representation

Identification and classification of the separation alternatives available to a given R-matrix is carried out by transformation into what is termed the S-matrix. The separ-

ation of two adjacent components, $j, j + 1$, in a given product, i is defined by :

$$S_{i,j} = \frac{r_{j,i}}{r_{j+1,i}} \quad (2.1)$$

Evaluation across the R-matrix provides the designer with information on the types of separations available; sharp, semi-sharp and non-sharp. The quantities, $S_{i,j}$, may change as the operations of separation and bypassing are performed, giving some information on the effect of a particular decision on the synthesis alternatives. The choice of separation tasks is therefore problem-specific. A particular task is not imposed on a problem, rather it is selected using the information provided by the S-matrix.

Splitting alternatives are confined to bypassing to the final product blenders. No splitting or blending is carried out within the separation section due to the inability of the R-matrix to represent these operations.

2.3.3 Component Assignment Diagram

The component assignment diagram (CAD), Cheng and Liu [7], is a matrix-like representation of the MCP synthesis problem. The manner in which the problem is represented could be easily put in the form of a matrix but the authors include some extra detail in the form of the component lines which provide visual information on the component distribution in each product stream. Component flows are used to describe the product set. The *products* are ranked in terms of volatility - the reasoning being that a stream with a higher volatility is more likely to be a distillation top product than a stream with a lower volatility. This ranking defines a set of separations which produces a decision tree for synthesis. Recognising that the transformations contained in the tree may not represent feasible separations the authors have devised a recursive decomposition strategy for dealing with such cases. The result of the removal of any infeasible transformation is the creation of an extra outlet stream and a sharp or semi-sharp separation. An example CAD and the resulting decision tree are shown in figures 2.9 and 2.10 respectively.

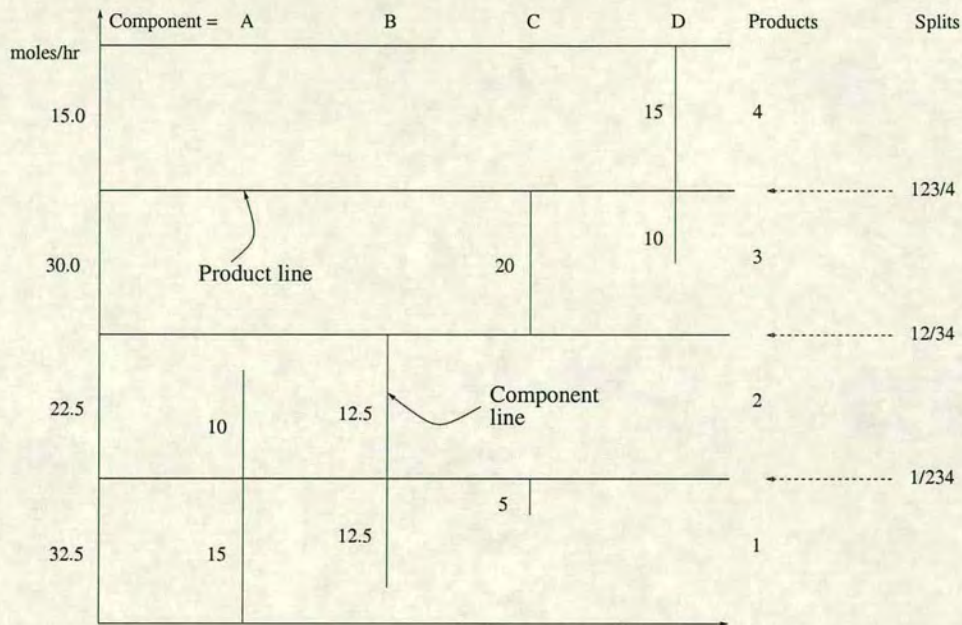


Figure 2.9: Example component assignment diagram

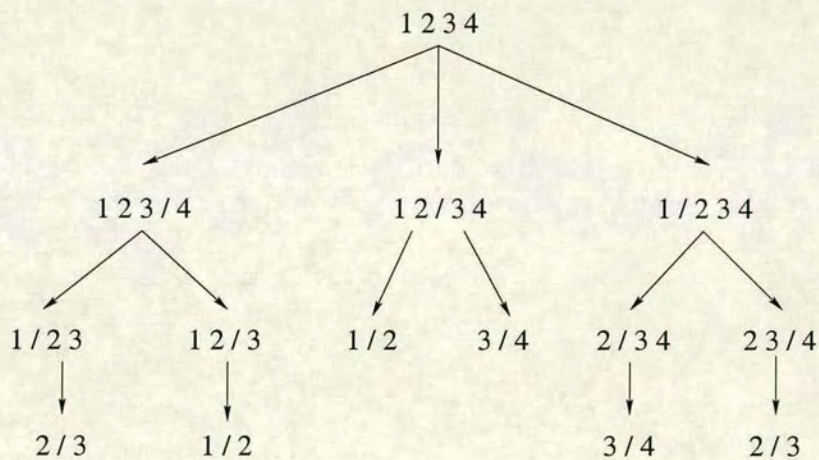


Figure 2.10: Decision tree for CAD example

Operations defined on the representation

The only operations which the decision tree defines are the separations and the final product blending which occurs due to product fragmentation associated with the removal of infeasible transformations.

Synthesis information drawn from the representation

The strength of this representation is that it shows the separation alternatives clearly. The separation tasks are problem specific: there is no *a priori* assignment of column functions nor is the number of separators restricted. The CAD is used to characterise the separation tasks by examining the distribution of components across a product line (see figure 2.9). The separation tasks are specified by the separator output streams. The degrees of freedom available to a distillation column limit the number of components which can be specified in those streams. To avoid this problem the authors permit only sharp and semi-sharp separations, which require only key component specifications. When nonsharp separations occur they are treated as infeasible cases.

Opportunities for bypassing are identified from the CAD by looking for *all-component-inclusive* products. In the example provided, product stream 3 is an all-component-inclusive product with respect to feed stream 34. A fraction of the feed can be bypassed to product 3. The authors select 90% of the maximum possible bypass fraction.

2.3.4 Stream Allocation Diagram

The Representation

The stream allocation diagram (SAD), Chen and Fan [6], is another matrix-graphical hybrid. Figure 2.11 illustrates a SAD representing a synthesis problem involving a feed stream \mathbf{f} , and the product streams, \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 .

These four streams are expressed as vectors as follows:

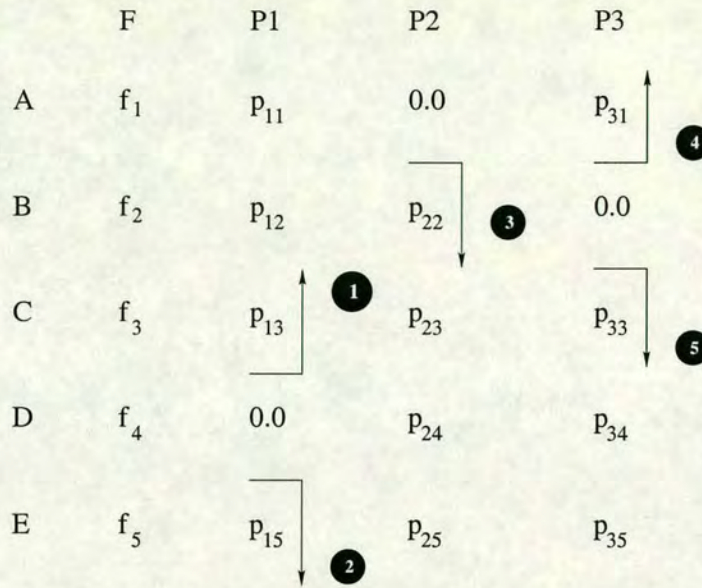


Figure 2.11: Example stream allocation diagram

$$\begin{aligned}
 \mathbf{f} &= [f_1 \ f_2 \ f_3 \ f_4 \ f_5]^T \\
 \mathbf{p}_1 &= [p_{11} \ p_{12} \ p_{13} \ p_{14} \ p_{15}]^T \\
 \mathbf{p}_2 &= [p_{11} \ p_{12} \ p_{13} \ p_{14} \ p_{15}]^T \\
 \mathbf{p}_3 &= [p_{11} \ p_{12} \ p_{13} \ p_{14} \ p_{15}]^T
 \end{aligned} \tag{2.2}$$

where elements $\{f_j\}$ and $\{p_{ij}\}$ represent the molar flow of the j th component, ranked in descending order of a certain physical or chemical property, in the feed stream and product stream i , respectively. In constructing the SAD, feed and product streams are listed as columns from the left to the right. For each product stream, a directional separation breakpoint (DSB) is defined in connection with any component in the product stream whose molar rate is zero. The DSB comprises a line and a numbered arrow; the line defines the key components of the separation and the arrow indicates which part of the product stream is to be matched by the feed stream.

Operations defined on the representation

This representation is manipulated on a stream-matching basis, i.e. parts of a given stream are matched with a particular product using bypassing, stream splitting and sharp separation. After any operation, the SAD is updated to show the remaining parts of the product stream which have yet to be satisfied. The current process stream under consideration occupies the feed stream column.

The identification of DSBs requires the presence of zero component molar flows in the stream allocation diagram. If none exist they can be created through maximum bypassing of the feed stream to each product - the amount bypassed to each product is subtracted from the both the feed and product columns of the SAD. Maximum bypassing will result in at least one zero component flow per product column.

Stream splitting and sharp separation are coupled by the following heuristic: *“Favour the stream splitting from which each of the resultant substreams, after undergoing one separation, matches with at least one product stream to the maximum extent”*. DSBs identify stream splitting opportunities required by that heuristic. Referring to figure 2.11, for DSB 1, part of the feed stream, $x_1\mathbf{f}$, where

$$x_1 = \min\left(\frac{p_{11}}{f_1}, \frac{p_{12}}{f_2}, \frac{p_{13}}{f_3}\right) \quad (2.3)$$

is first processed in separator with a breakpoint of 3 whose resulting top product is matched with product 1. The top product component flows are subtracted from product 1 in the SAD, indicating how much of that product has yet to be satisfied.

Synthesis information drawn from the representation

The SAD contains alternative sets of DSBs which provide the stream splitting and associated separation alternatives available at a given point during synthesis. The splitting heuristic serves to identify a small but useful set of splitting alternatives. However, the generation of DSBs is dependent on maximum bypassing to product blenders at every

opportunity, resulting in exclusively sharp separation alternatives. For cases where the initial problem does not have any zero component flows in each product, maximum bypassing will create these.

2.3.5 Separation Product Matrix

The Representation

This separation product matrix (SPM) of Liu and Xu [21–23] is a matrix-based representation of the multicomponent product separation problem. Components are ranked by volatility and it extends the R-matrix representation of Bamopoulos *et al.* [4]. A k -component, m -feed and n -product separation process is represented as follows:

$$\begin{bmatrix} \frac{f'_{1,1}}{fs_{1,1}} & \dots & \frac{f'_{1,k}}{fs_{k,k}} \\ \dots & \dots & \dots \\ \frac{f'_{m,1}}{fs_{1,1}} & \dots & \frac{f'_{m,k}}{fs_{k,k}} \end{bmatrix} \begin{bmatrix} fs_{1,1} & & \\ & \ddots & \\ & & fs_{k,k} \end{bmatrix} \begin{bmatrix} A_1 \\ \vdots \\ A_k \end{bmatrix} \rightarrow \begin{bmatrix} \frac{p'_{1,1}}{ps_{1,1}} & \dots & \frac{p'_{1,k}}{ps_{k,k}} \\ \dots & \dots & \dots \\ \frac{p'_{n,1}}{ps_{1,1}} & \dots & \frac{p'_{n,k}}{ps_{k,k}} \end{bmatrix} \begin{bmatrix} ps_{1,1} & & \\ & \ddots & \\ & & ps_{k,k} \end{bmatrix} \begin{bmatrix} A_1 \\ \vdots \\ A_k \end{bmatrix} \quad (2.4)$$

where

$$fs_{j,j} = \sum_i f'_{i,j} \quad (fs_{r,j} = 0, \quad \text{if } r \neq j) \quad (2.5)$$

and

$$ps_{j,j} = \sum_i p'_{i,j} \quad (ps_{r,j} = 0, \quad \text{if } r \neq j) \quad (2.6)$$

$f'_{i,j}$ is the flow rate of component j in feed i

$p'_{i,j}$ is the flow rate of component j in product i

A is the component vector - the authors have modelled the separation process as an algebraic transformation on this vector.

Equation 2.4 can be denoted as :

$$\mathbf{F} \quad \mathbf{FS} \quad \mathbf{A} \quad \rightarrow \quad \mathbf{P} \quad \mathbf{PS} \quad \mathbf{A} \quad (2.7)$$

where \mathbf{F} is called the feed matrix. The elements of \mathbf{F} are:

$$f_{i,j} = \frac{f'_{i,j}}{fs_{j,j}} \quad (2.8)$$

and \mathbf{P} is called the Separation Product Matrix (SPM), the elements of which are:

$$p_{i,j} = p'_{i,j}/ps_{i,j} \quad (2.9)$$

The separation product matrix is the transposed version of the R-matrix proposed by Bamopoulos *et al.* [4]. For systems where no reaction occurs \mathbf{FS} and \mathbf{PS} are the same and equation 2.7 reduces to:

$$\mathbf{F} \quad \mathbf{A} \quad \rightarrow \quad \mathbf{P} \quad \mathbf{A} \quad (2.10)$$

For a separation problem with multiple feeds, an Individual Feed Recovery Matrix (IFRM) is proposed to represent the relation between the products and each feed. The elements of IFRM, \mathbf{R} , for feed g and the products are:

$$r_{i,j}^g = p_{i,j}/f_{g,i} \quad (2.11)$$

Some elements of the IFRM are greater than unity. For a single feed problem the IFRM and the SPM are the one and the same.

Operations defined on the representation

Sharp separation is defined by cleaving a given IFRM or SPM along the columns between the adjacent key components for that separation.

Bypassing a fraction, $x_{i,j}$ of feed i to product j is defined on the feed matrix, \mathbf{F} , by:

$$\mathbf{F} - \begin{bmatrix} (\sum_j x_{1,j}) \mathbf{f}_1 \\ \cdot \\ (\sum_j x_{m,j}) \mathbf{f}_m \end{bmatrix} \quad (2.12)$$

and simultaneously on the product matrix, \mathbf{P} , by:

$$\mathbf{P} = \begin{bmatrix} (\sum_i x_{i,1}) \mathbf{P}_1 \\ \vdots \\ (\sum_i x_{i,n}) \mathbf{P}_n \end{bmatrix} \quad (2.13)$$

Finally, the authors do not explicitly define the operation of internal blending but do make use of it during synthesis. It consists of adding the appropriate elements of a set of feed or SPM matrices together.

Synthesis information drawn from the representation

Sharp separation and maximum bypassing of streams to the products go hand-in-hand. Since the sharpness of separation cannot increase, maximum bypassing lowers the mass load on the process separators thus decreasing the cost of separation. The maximum bypass from feed, g , to product i is:

$$b_{i,max}^g = \min_j r_{i,j}^g \quad (2.14)$$

where $r_{i,j}^g$ is defined by equation (2.11). After this operation, the feed matrix and IFRMs are updated and separation carried out. After separation, the matrices are checked for more bypassing opportunities. Where these opportunities arise, the possible maximum bypassing is always carried out.

For single feed problems the authors have identified the minimum required flowrate for a given separator (defined by its key components) to meet the the process separation requirements. This gives rise to a particular split fraction for the current problem which does not correspond to the maximum bypass fraction at that point, i.e. another splitting alternative is identified.

Three opportunities for internal blending have been identified by Liu and Xu [21–23]:

1. When two or more streams of the same composition appear, then blending of those streams is performed.
2. If the minimum flow of key components for a separation is not met, then streams are blended such that this criterion is satisfied.
3. There is a limit on the number of separators allowed in the process. When no bypassing alternatives exist and there are not enough separators left to operate on all the remaining streams, then blending of streams to reduce their number is undertaken.

Of all the methods described, [21–23] get the most out of their problem representation. Only sharp separations are allowed but opportunities are identified for stream splitting, bypassing and internal blending.

2.3.6 Superstructure

A superstructure is needed for mixed-integer-linear-programming (MILP) and mixed-integer-nonlinear-programming (MINLP) solutions to the MCP problem. The superstructure embeds “all possible” alternatives. The actual superstructure is dependent on the assumptions made by the designer. For example, Aggarwal and Floudas [2] and Wehe and Westerberg [45] assign column functions while Smith and Pantelides [36] do not. For their given assumptions the superstructures contain all structural alternatives, but each is different. This representation requires a lot of work on the part of the designer who must define the superstructure. Also, superstructures are neither general enough nor specific enough to encompass different problems effectively. They are not general in the sense that if eg., an extra component is added to the problem, the superstructure may have to be reformulated and not specific in that a superstructure may contain many infeasible alternatives for a given problem.

2.3.7 Summary and Conclusions

The features of each of the representations described are summarised in table 2.3.

	MAD	mMAD	R-matrix	CAD	SAD	SPM	Superstructure
Number of feeds	1	1	1	1	1	≥ 1	≥ 1
Number of products	2	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2	≥ 2
Separation types	S,SS	S,SS	S,SS	S,SS	S	S	S,SS,NS
Bypassing	yes	yes	yes	yes	yes	yes	yes
Splitting	yes	yes	no	no	yes	yes	yes
Internal blending	yes	yes	no	no	no	yes	yes

Table 2.3: Representation features

where S = sharp separation, SS= semisharp separation and NS = nonsharp separation.

There are a number of clear deficiencies in the existing representations, notably that most cannot handle problems involving more than a single feed and only the superstructure-programming formulation offers the ability to consider nonsharp separations. With the exception of nonlinear programming formulations, all product specifications must be in the form of point compositions, i.e. a single point in composition space, since the construction and use of the problem representations depend on this. As a result, it will be difficult to make use of effect product specifications.

2.4 Evaluation Problem

“Can the alternatives be evaluated effectively so that they may be compared ? Effective evaluation means a balance between speed and accuracy.” Nishida *et al.* [30]

To evaluate an alternative is to attach some measure of how good the alternative is to it. The measure is usually cost or cost-related, but could be in terms of other process considerations such as safety or controllability. Evaluation can be carried out through the use of models to design and/or cost the process units or by a measure linked to cost, eg. the relative volatility of separation key components. The effort expended on the evaluation of synthesis alternatives is dependent on the desired accuracy of the

solutions which is in turn dependent on the design stage.

In the early design stages, the aim is to generate sets of feasible (hopefully good) solutions. Perhaps of more importance is the reduction of the search space through the elimination of infeasible alternatives and the worst solutions. Given the initial large size of the search space and the emphasis on feasibility rather than rigour, this calls for the use of fast shortcut models and costings. For the detailed design stage, rigour is desired and the models which provide this are employed.

Most of existing MCP synthesis methods are intended for use in early design. The shortcut models and costings which have appeared are described briefly.

- Muraki *et al.* [27] and Floudas [11] use the following as a means to cost distillation columns.

$$Cost_i = (L_i D_i)^{(0.6)} \quad (2.15)$$

where L_i and D_i are, respectively, the separation mass load (the molar flowrate through the column) and the difficulty of the i th separation. This costing is only intended for sharp separations.

- The cost correlations of Rathore *et al.* [33] are used by Bamopoulos and coworkers [4], Cheng and Liu [7] and Muraki and Hayakawa [28]. These are used in conjunction with the Fenske-Underwood-Gilliland shortcut distillation column design which is intended for sharp and semi-sharp separations.
- Wehe and Westerberg [45] and Chen and Fan [6] obtain column costs using a fixed charge model based on material flow through the column.

$$Cost_i = (f_i d_i) + c_i \quad (2.16)$$

where f_i is the flow through column i , d_i is the difficulty of the i th separation and c_i is the coefficient associated with the fixed cost for separator i . Again, this model is intended for sharp separations.

The models used are consistent with the requirements for early design, i.e. they provide quick estimates of the cost for a given separation task.

2.5 Synthesis Strategy

“The aim of any synthesis strategy is to locate the best solutions without enumerating every alternative.” Nishida *et al.* [30]

Synthesis strategy involves the selection of a suitable search method for a synthesis problem. The range of approaches available for this purpose was discussed in section 1.1 under the headings of heuristic, evolutionary and algorithmic. The various other features of concern relate to the suitability of the method for early design, i.e. its ability to return a set of solutions, whether alternatives are generated before or during synthesis, the effectiveness of the strategy in eliminating infeasible or poor solutions and the ease of use of the chosen strategy.

2.5.1 Heuristic Strategies

Bamopoulos *et al.* [4] employ a two-stage synthesis method. The generation of alternatives and the search for the best alternative are not done concurrently. Starting from the feed set, alternative generation is carried out by identification and inclusion of the separation and bypassing opportunities from the R-matrix (section 2.3.2) for each sub-problem encountered, up to a prespecified number of separation units. This creates a tree representing a set of feasible alternative flowsheets, for which the separation tasks and the flow and composition of each stream are known.

The second stage of synthesis performs the evaluation of and search for the best alternatives. The tree is searched using a best-first search. This expands the partial sequence that is the most promising and abandons it, in favour of another as soon as things look bad compared to the current best solution. Taking into account the approximate nature of the costings, the authors retain alternatives costing up to 30% more than the best solution. This is an effective approach for early design generating both the structural and design alternatives during synthesis. However, the search method employed does not give the designer full control over the size of the solution set. This is dependent on the cost bound about the cost of the best solution.

Cheng and Liu [7] also cast the synthesis problem in the form of a decision tree, derived from their CAD (section 2.3.3) problem representation. Only feasible separation sequences are described by the decision tree, which does not consider nonsharp separations. Bypassing opportunities to final products are always taken to 90% of the maximum amount possible. Established sharp separation heuristics are used to locate the best alternatives which are then designed and costed. This strategy is consistent with the separations which occur in the decision tree but cannot reliably be extended to nonsharp separations if they are also considered. However, the synthesis strategy need not rely on heuristics as the decision tree is amenable to an algorithmic search method.

A synthesis procedure with special emphasis on stream splitting is presented by Chen *et al.* [6]. Synthesis is guided by the heuristic - "Favour the stream splitting from which each of the resultant substreams, after undergoing one separation, matches at least one product stream to the maximum extent possible". Sharp separation is assumed, which facilitates the heuristic and reduces the number of alternatives to be considered. A one-step, look-ahead technique is applied which detects infeasible stream splitting arrangements at an early stage of synthesis. The effect of the heuristic is to maximise the bypassing of separators and to favour parallel structures. The method locates the alternatives which possess feasible stream splitting arrangements. The authors are concerned only with alternative generation and therefore no consideration is given to searching for the best solutions to a given problem.

2.5.2 Evolutionary Strategies

Muraki *et al* [27–29] propose a two-stage strategy for the synthesis of multicomponent product processes - a search for the optimal separation sequence followed by optimisation of that process through the introduction of stream splitting and blending. The search for the optimal separation task sequence is carried out using an evolutionary method for the synthesis of pure component products with an initial separation sequence generated from the rule that difficult separations are carried out last. The separation tasks are defined by pairs of adjacent key components and only one in-

stance of each keys split is permitted in any flowsheet. The evolutionary rule then develops the alternative separation sequences by interchanging the relative positions of two neighbouring separators. The second synthesis stage searches for the optimal stream split fractions within the current separation sequence by varying the amount separated in each separator. The problem representations, MAD and mMAD (section 2.3.1), provide alternatives for the values of those stream splits and the means to test the feasibility of the resulting flowsheets. This involves repeated manual manipulation of the diagrams. Synthesis is terminated when the value of the objective function is not improved by the application of the evolutionary rule. The original method for MCP synthesis allows only for sharp separators. When semi-sharp separators are introduced the two-stage method is retained but the second stage also involves the optimisation of the degrees of separation sharpness, i.e. the recoveries of the key components in each separator output in the current sequence. A set of linear equations solved to determine the stream division ratios for given degrees of separation sharpness.

Though this method is capable of generating a set of solutions to a given problem its requirement to manually test the feasibility of flowsheets makes it time-consuming and cumbersome, reducing its usefulness as an early design method.

2.5.3 Algorithmic Strategies

A superstructure which embeds all possible alternatives is used in the formulation of a MINLP by Aggarwal and Floudas [2] and Floudas [11]. Smith and Pantelides [36] use a superstructure to formulate an NLP. The assumptions made for each superstructure are discussed in section 2.3.6. In each case the resulting problem is solved for the optimal flowsheet structure and design variable values. These approaches are particularly cumbersome for the earliest design stages as they require the structural alternatives to be provided by the user before synthesis in the form of a superstructure, whose construction is time-consuming. Also, the search methods return only a single solution and this solution is not guaranteed to be optimal. For early design the ability to return a set of solutions is crucial.

Wehe and Westerberg [45] use a superstructure to formulate a linear programming problem. All models used, including the cost model are linear except for the splitters. Because of the assumption of sharp splits, it is possible to analyse the splitters involved in each alternative to provide a linear model for every splitter present in a three component problem. A linear program can be formulated and solved for each of the two possible sequences. The global optimum is the less expensive of the two. For problems with greater than three components, the splitter constraints are relaxed to a linear form. For each structural alternative a linear program can be formulated and solved, providing a lower bound on the corresponding nonlinear program. Solving the nonlinear program with the most promising structure then supplies an upper bound for the solution to the problem. The alternative structures whose lower bound is greater than that upper bound are discarded. The nonlinear programs (NLPs) for the remainder are then solved to yield their upper bounds. Upper and lower bounds within 1% of each other strongly imply global optimality.

Again, this method requires that the designer provides a superstructure before synthesis and only returns a single solution. The solutions obtained for 3 component problems are guaranteed to be optimal while those for problems with greater than 3 components have a reasonable guarantee of optimality. However, to provide that guarantee, a linear cost function must be used and this shortcut costing is at odds with a claim of rigorous optimality, i.e. the re-costed flowsheet using rigorous nonlinear models and costs cannot be guaranteed to be optimal.

2.5.4 Synthesis strategies - summary and conclusions

The features of the synthesis strategies described are outlined in table 2.4. The applicability of each strategy to the early design stage is then assessed.

Of the synthesis strategies described, that of Bamopoulos *et al.* [4] is truest to the aims of early design, i.e. it generates the alternatives during synthesis, returns a set of solutions and does so in a clear, straightforward manner. The two-stage approach works well, first generating a tree of feasible flowsheet alternatives using heuristics and

Method	Generation of alternatives	Set of solutions	Search type
Bamopoulos <i>et al</i>	during search	yes	heuristic
Cheng and Liu	during search	yes	heuristic
Chen and Fan	-	no	none
Muraki <i>et al</i>	during search	yes	evolutionary
Aggarwal and Floudas	before search	no	algorithmic
Floudas	before search	no	algorithmic
Smith and Pantelides	before search	no	algorithmic
Wehe and Westerberg	before search	no	algorithmic

Table 2.4: Synthesis strategy features

then determining the best solutions using an algorithmic search method. However, the authors' limiting assumptions of sharp and semi-sharp separation and exclusion of stream splitting and blending reduce the number of alternatives available. Cheng and Liu [7] pass up the opportunity to apply an algorithmic search method to a well-defined decision tree, choosing instead to use sharp separation heuristics to locate the best solutions. Their assumption of near maximum (90%) bypassing which results in sharp/nearly sharp separations ensures that these heuristics can be used but that assumption could have been relaxed if an algorithmic search method had been applied. The evolutionary and algorithmic strategies described do not perform well as early design methods. The method of Muraki *et al* [27–29] requires much trial and error on the part of the designer resulting in a time-consuming design process while the superstructure-programming procedures need to be supplied with the structural alternatives in the form of a superstructure and return only a single solution which is not guaranteed to be the global optimum.

2.6 Conclusions

From discussing the various features of the synthesis procedures proposed in the literature the attributes of a good, early design stage approach to MCP synthesis problems have can be described. The ideal synthesis method would have the following features :

Effect product specifications

Effect products form an important class of multicomponent products. However, many

of the synthesis methods described deal solely with point composition specifications and depend on them to formulate the problem representation. An MCP synthesis method which can handle any product specifications, regardless of the stream properties involved, is desired.

Set of solutions

At the early design stage it is essential to be able to synthesize a set of good solutions to the given problem. A single optimum solution is not likely to satisfy factors such as controllability, safety and environmental acceptability. Nor it is likely that the solution will remain the optimum when recosted using more rigorous costs.

Algorithmic search method

Experience of processes involving nonsharp separation, stream splitting and blending is very limited. Heuristic and evolutionary rules rely on the use of knowledge or the existence of current processes, greatly limiting the scope of possible solutions. Therefore, an algorithmic search method is required as they are suitable for use in the development or design of new processes. Furthermore, an algorithmic approach is amenable for use in the automated generation of processes which is desirable in the early stages of design.

Generation of structural alternatives during synthesis

It is desirable to generate the structural alternatives as synthesis progresses as this reduces the workload on the designer. However, a rich search space must still be retained. The literature methods that build up the process structure in this manner do so by adding units when the opportunity for their use is detected. This opportunity also fixes the unit design since the unit is chosen to perform a particular task. The detection of such opportunities is in turn dependent on the limiting assumptions used by the methods, eg. sharp separation, maximum bypassing to product blenders etc.. A much richer set of structural alternatives is needed.

Non-sharp separations

This is a non-sharp problem: therefore, it follows that one should consider non-sharp separations in its solution. Most procedures in the literature allow for sharp or semi-sharp separations. This simplifies the problem, but in so doing reduces the number

and quality of the alternatives available.

No *a priori* assignment of column functions

This is related to the last point. In some approaches the number and functions of columns in a flowsheet have been decided before the synthesis step. For example, Muraki and Hayakawa [27], assign each column a different pair of key components and only one instance of a pair is allowed. Column functions are facilitated by the assumption of sharp or semi-sharp separations. Bamopoulos *et al* [4], Cheng and Liu [7] avoid this assignment but still use only sharp and semi-sharp separations. Smith and Pantelides [36] are the only workers who avoid specifying column functions and make nonsharp separations available.

Inclusion of stream splitting and blending

In MCP synthesis stream splitting and blending must be included. The presence of more than one of the feed components in the product streams provides opportunities for these operations which should not be ignored.

In the next chapter a synthesis method for MCP problems which meets these requirements will be presented.

Chapter 3

MCP Synthesis Method

3.1 Introduction

A synthesis method for MCP separation problems will now be presented. The problem state and stream state definitions which play an important role in the work are first described. This is followed by the development of the structural alternative generation algorithm and the unit design alternatives. The resulting search space is examined and the algorithm selected to explore it is described. Finally, a more detailed examination of the search space is carried out to identify and remove repetitions of process alternatives and to assess the range of structures provided by the synthesis method.

3.2 State definitions

The following two definitions are central to the development of the synthesis method. The *problem state* is one component of the synthesis problem representation while the *stream state* plays an important part in the derivation of the operations on the problem state.

Problem state

Process synthesis, however performed, is an iterative procedure. In the search for a solution, all methods repeatedly apply a set of operations to a given problem. Each iteration takes the current problem as input and operates on it. The output of a successful operation then becomes the problem. It is necessary to describe the problem at any point during synthesis. This is the problem state and should contain the following:

- The inputs which make up the problem. For a separation synthesis problem these are the streams requiring separation and the targets which that separation must satisfy.
- The definition of the solution state. The synthesis method must devise the means to identify a solution.

For this work the synthesis problem is described by the process stream front (PSF) and a set of targets which a solution must satisfy. The former is the set of streams which do not yet satisfy the given targets. These targets may be product stream specifications, environmental or controllability requirements or other process constraints which a designer wishes to impose. At the start of synthesis the PSF consists of the feed stream set. Process units are added which consume and produce streams thus altering the process stream front and hence the synthesis problem. It is also possible to change the synthesis problem by adding, removing or updating targets. A solution to a given problem has been obtained when the process stream front is empty and there are no remaining targets to be satisfied, i.e. no streams remain and all specifications have been met.

Stream state

The definition of the stream state is important to this work. The state is comprised of two parts:

- The *intensive stream state* describes the thermodynamic state of a stream, that part of the stream which is independent of flow. By Gibbs phase rule [38] $nc + 1$ independent variables are required, where nc is the number of components in the stream. For example, temperature, pressure and $nc - 1$ mole fractions is one possible variable set. As long as the variables used to describe the intensive stream state are independent, values for all other thermodynamic quantities (eg. phase, enthalpy etc.) can be determined.
- The *extensive stream state* is that part of the stream description which is dependent on the flow of a stream. Only a single variable need be used.

3.3 Structural alternative generation

The automatic synthesis of process flowsheets is most desirable at the early design stage. This is the point in the design process where the most alternatives arise. The designer's effort is best spent in the selection of the flowsheets for the next stage of synthesis rather than in their generation.

Process flowsheets consist of two distinct sets of variables. Firstly, there exist the discrete structural variables which control the selection and connection of units in a flowsheet. The other set is comprised of the variables (which may be discrete or continuous) which describe the unit and stream states. Automatic synthesis of process flowsheets requires the automatic generation of alternatives for both sets of variables.

The MCP synthesis methods described previously (Chapter 2) consider a range of structural alternatives. In some cases, the generation of these alternatives occurs as synthesis progresses and is amenable to automation. However, in many cases, the selection and addition of a unit is dictated by the opportunity for use of that unit - an opportunity which also fixes the unit design since the unit is chosen to perform a particular task. The detection of such opportunities is in turn dependent on the limiting assumptions used by the methods, eg. sharp separation, maximum bypassing to product blenders etc.. It is desirable to generate a much richer set of structural alternatives.

Friedler, Fan and coworkers [14], [15] are developing methods for the automatic creation of superstructures. However, this work does not meet our requirements because the unit functions must be assigned beforehand in order to create the superstructure. These functions are expressed in terms of the components present in the unit input and output streams. This is not useful in characterising nonsharp separations. Smith and Pantelides [37] cover all structural alternatives by assuming complete connectivity between a set of process units. However, this creates a very large problem and embeds many infeasible structures.

The desired features of a method for generating structural alternatives are as follows:

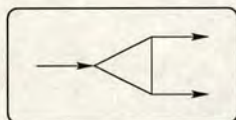
- It must encode a rich set of structural alternatives. For the early design stage, as many alternatives as possible must be considered.
- It must make systematic use of splitters, blenders and separators. There must be a *reason* for the appearance of each of these at a given point in a structure. The aim is to try to avoid the generation of clearly infeasible or redundant structures such as long chains of splitters and/or blenders.
- It is desirable to encode the alternatives implicitly, i.e. they unfold as synthesis progresses. This has the very important benefit of not having to describe all structural alternatives before synthesis.

A structure generation method incorporating the features described is developed through first providing a framework for analysing unit functions. This results in a set of rules for the addition of process units which is then formulated as a set of operations on the process stream front. Finally, an algorithm for the generation of the structural alternatives is presented.

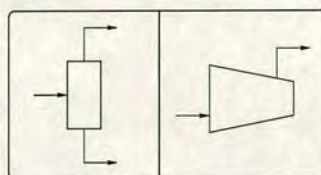
3.3.1 The framework for structure generation

The aim is to generate process structures in both an implicit and systematic fashion. A possibility is to try processing each stream created during synthesis with each unit

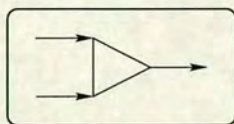
type available. However, this will generate many redundant or infeasible structures and create a very large search space. To overcome this the units must be assigned a particular task. For this work, unit tasks are defined in very basic terms - the manner in which they contribute to stream intensive state diversity. The more *different* stream states created, the greater the exploration of stream space and the more possible paths to a solution. The intensive stream state is used as it is the appropriate measure of diversity because an infinite number of different extensive stream states and hence full stream states could be created by splitting a stream an infinite number of times. This is not a reflection on the extent of exploration of stream space. The units available to an MCP process are splitters, blenders and *central transformation units*. Their functions are examined with respect to their contribution to stream intensive state diversity.

Stream splitter

A splitter divides the input stream into two or more output streams of equal composition but differing flow. It alters the input extensive stream state and creates *copies* of the input intensive stream state. Splitters provide a means of distributing a given intensive stream state about a flowsheet. To consider more than one destination for any intensive stream state, i.e. to make that state available to more than one unit, the stream must be operated on by a splitter.

Central transformation unit

A central transformation (CT) unit, eg. separator, compressor etc., does work on a stream or set of streams to create a new set of stream states. CT units have degrees of freedom over and above their input streams, i.e. their outputs are not fully determined by their inputs. Therefore, they provide the potential to reach new intensive stream states not yet encountered in the process. To get the most out of this potential, CT units must operate on each intensive stream state created.

Stream blender

A blender creates a single output from multiple input streams, bringing about a change in both intensive and extensive states. Since its output stream is fully determined by its inputs, a blender's usefulness lies in the ability to access new stream states bounded by the CT unit outputs and the feed stream set.

3.3.2 Rules for the addition of process units

Having established the unit functions with respect to stream state diversity, two rules are formulated for the addition of units within a flowsheet such that those functions are preserved.

Rule 1

The outputs of a splitter must each have a different destination, i.e. be inputs to different units. This rule has two results. Firstly, it prevents the recombination of outputs from the same splitter but does not prevent blending of identical intensive stream states arrived at by different routes. Secondly, it does not permit a given CT unit to process more than one copy of the same intensive stream state. The decision to introduce a given intensive stream state at multiple points in a unit, eg. as in tubular reactor feed, is considered to be the domain of the unit design.

Rule 2

No more than two copies of the same intensive stream state need appear in the process stream front. It is the intention that the flowsheet will be built up as synthesis progresses. Therefore, only one destination for an intensive stream state will be considered at any time. A second copy is maintained, allowing it to be processed by a different route to the first - but only after the first copy has been consumed by a unit and is no longer part of the process stream front. The result of this rule is that the splitters responsible for setting the inputs to CT units and to blenders need only have two outputs.

The two rules combined require that two copies of a given intensive stream state are maintained in the process stream front - this is the minimum number of copies required to consider different destinations for that stream state. Lone or single intensive stream states in the PSF must be split into two fractions which will be processed by two different paths. However, the rules possess symmetry, i.e. each of the two copies of an intensive stream state could go to either destination. As this would result in the same structures being generated more than once, the symmetry must be broken. This is achieved by introducing *types* of streams whose function is to select one of the intensive stream state copies for the current destination. Three types are assigned to process streams as follows:

- A *newstream* (NS) is any new intensive stream state that appears in the process. It is a lone intensive stream state and is split into two fractions - the *immediate gratification* fraction and the *loiter with intent* fraction.
- The *immediate gratification* (IG) stream fraction is the stream intensive state copy whose destination is set first.
- The destination of the *loiter with intent* (LWI) stream fraction is not assigned until after that of its IG sibling is set. In other words it is not processed until its IG sibling has been consumed by a unit and is no longer part of the process stream front. At this point the LWI stream is a lone intensive stream state. It will be split again to generate another IG-LWI pair.

Applying the stream types and processing rules

The stream types and processing rules control the addition of units to a process. This is illustrated by building up a process structure starting from a single feed and using a simple (1-input, 2-output) separator as the CT unit.

1. The feed stream is a newstream. It is split into its IG and LWI fractions (figure 3.1).

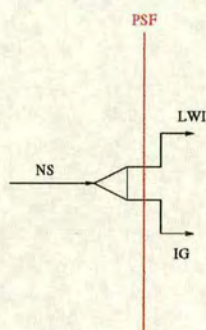


Figure 3.1: Splitting the feed stream

2. Two copies of the feed intensive stream state exist in the process stream front. As this is the maximum number of copies needed, no more splitting of these streams is carried out at this point. Blending is not permitted as each copy must have a different destination. The only available unit type is the CT unit. Therefore, the IG fraction is processed using the separator (figure 3.2).

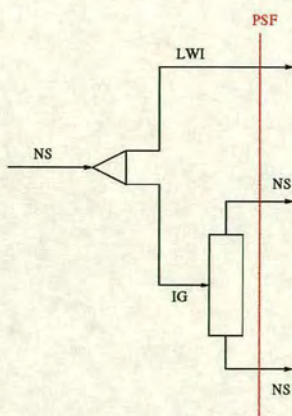


Figure 3.2: Adding the separator

3. Blending could be performed at this point to access new stream states. However, this does not exploit the full potential of stream state diversity. The process stream front is composed of an LWI stream and two newstreams. All are single copies of an intensive stream state. Each is split to generate IG-LWI pairs. (figure 3.3).

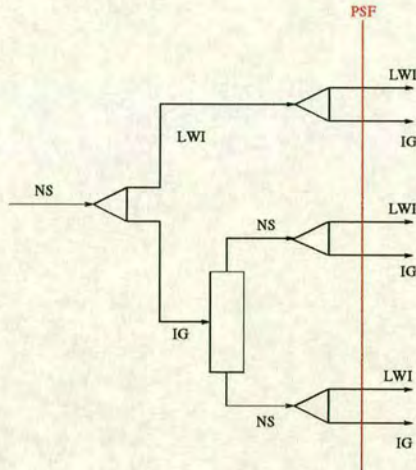


Figure 3.3: Splitting the lone intensive stream states

4. Blenders can now exploit the opportunities to access new stream states bounded by the separator output and feed stream states. To avoid wasting separation work the separator output streams are not mixed. Each separator output IG fraction then forms an input to a different blender and the process feed IG fraction is distributed between these. In general, the appearance of blenders in the flowsheet corresponds to the appearance of separator output streams.

The four steps described form a repeatable group. *All* the outputs from the last step are single intensive stream states. These will be split into IG-LWI pairs. This will be followed by the addition of a separator whose outputs will then undergo an IG-LWI split before the blending of streams occurs.

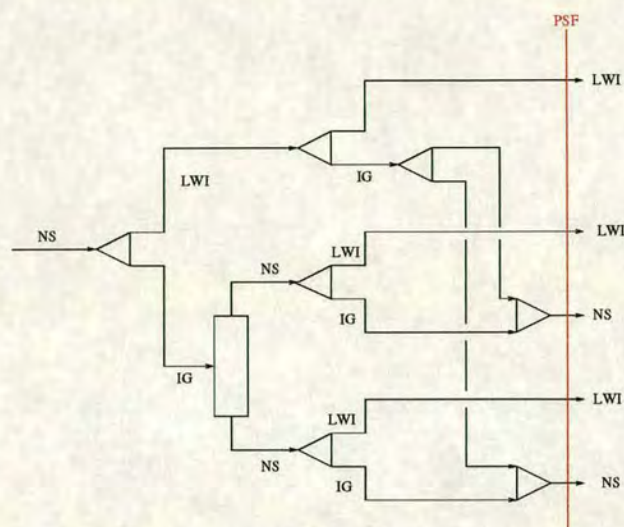


Figure 3.4: Taking the blending opportunity

3.3.3 Defining structure generation operations

The synthesis problem representation is comprised of the problem state description and the allowed operations on it. The problem state has been defined - it is the process stream front and the set of targets which a solution must satisfy. The structure generation operations are required to alter the problem in a defined or controlled manner. Furthermore, in process synthesis, these changes can only be brought about by the addition of process units. The application of the stream types and processing rules fulfill these requirements - they introduce process units in an ordered manner. The resultant sequence of four steps will now be exploited for the definition of operations on the synthesis problem.

The four steps give rise to three different operations which are each responsible for the addition of specific units to the process flowsheet and for the management of the associated alternatives. Firstly, there is the *IG-LWI split* which splits lone intensive stream states into IG-LWI pairs. This is followed by the *central transformation* operation which controls the addition of central transformation units. The *internal blending* operation manages the blending of streams within the flowsheet. Finally, an operation not derived from the processing rules is presented: the *target acceptance branching* operation considers whether or not to accept streams which satisfy target specifications

as products. The description of each operation is broken down into the following components:

- The *inputs* are the streams and units required by the operation.
- *Failure cases* are those inputs which render an operation infeasible, i.e. it cannot function and produces no output. If the current operation is infeasible all subsequent operations are also infeasible since they require an output from previous operations. Failure cases result in the termination of structure generation for the inputs under consideration.
- Each operation (with the exception of target acceptance branching) is responsible for adding units to the flowsheet. The *structure created* by each operation and its dependency on the operation inputs is described.
- The *structural alternatives* available to each operation are outlined.

Immediate-gratification/loiter-with-intent split (IG-LWI split)

This operation is responsible for making different destinations available to each intensive stream state by maintaining two copies of each stream state in the process stream front. This is achieved by splitting lone intensive stream states into IG-LWI pairs.

Inputs	The streams involved in the operation are the lone intensive stream states in the process stream front, i.e. newstreams and LWI streams whose IG siblings have been consumed.
Failure cases	There are no failure cases. This operation is always successful.
Structure created	Each stream involved in this operation forms the input to a binary (two-output) splitter.
Structural alternatives	There is only the single structural alternative, i.e. the set of binary splitters, one for each input stream to this operation.

Central Transformation (CT)

For this work, it is assumed that splitting and blending alone cannot generate the product set, i.e. that central transformation units are required. Therefore, the central transformation operation is of key importance in the generation of structural alternatives. Its function is to control the addition of CT units to the process flowsheet and without it solution structures cannot be obtained. It is the only operation with failure cases, i.e. for certain inputs a CT unit cannot be added. When this occurs no other units will be added to the process, since it is no longer possible to reach new stream states.

Inputs	The inputs to the operation are the IG streams in the current process stream front and the available CT units.
Failure cases	<p>The operation fails when there are no input streams to it and hence no inputs to the available CT units.</p> <p>Another failure occurs when there are no more CT units left, i.e. all the available CT units have been used.</p> <p>Failure also occurs if the number of input streams to the operation does not match the number of inputs to the selected CT unit type, eg. a single feed column cannot process more than one feed stream.</p>
Structure created	An instance of a central transformation unit is created. The number of input and output streams depends on the type of unit selected.
Structural alternatives	More than one type of CT unit can be made available to this operation. Therefore, a choice of CT unit types exists.

Internal Blending (IB)

The function of this operation is to control the stream blending which allows processes to reach stream states within the bounds set up by the feed set and the outputs of CT units.

Inputs	The allowed inputs to the operation are the IG streams in the current process stream front.
Failure cases	There are no failure cases for this operation. If no inputs exist, then the operation does not create any units and has no effect on the process stream front.
Structure created	Instances of blenders correspond to the number of newstreams created by the last CT unit added to the flowsheet. These newstreams will have been split into IG-LWI pairs with the IG fractions each giving rise to a blender. The remaining IG inputs to the operation are distributed between the blenders. The number of outputs from the splitters used to perform this distribution is equal to the number of blenders, i.e. the number of destinations available. Figure 3.5 illustrates.
Structural alternatives	The restriction that streams created by the same CT unit cannot be mixed results in mutually exclusive input sets to the blenders. The IG streams created by the last CT unit are not affected by this, as they are kept separate from the outset. It is the streams which are distributed between the blenders which may be partitioned.

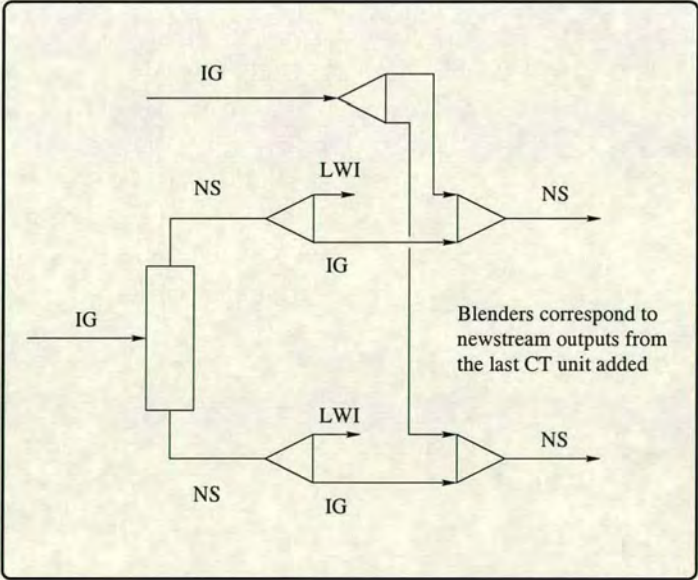


Figure 3.5: Internal blending structure

Target acceptance branching (TAB)

For MCP problems, different streams may satisfy the same specification and should therefore be blended. Alternatively, an off-specification stream blended with one which satisfies the specification may still result in an acceptable product. In order to capture such cases, a choice has to be made whether or not to accept a stream which satisfies a set of specifications as a product. In other words, to remove it from or keep it in the process stream front. The target acceptance branching operation is introduced to manage this alternative.

Inputs	The inputs to the operation are streams which have satisfied product specifications.
Failure cases	There are no failure cases for this operation. If there are no inputs then the operation has no effect on the process stream front.
Structure created	The operation does not create any units. Its effect is either to remove or to retain the input streams in the process stream front.
Structural alternatives	The alternatives considered are in accordance with the function of this operation, i.e. to accept a stream as a product or not.

3.3.4 Structure generation algorithm

The algorithm for structure generation is a cycle of operations which builds up the flow-sheet structure with each iteration. Implicit enumeration requires that the algorithm be driven purely from the problem state, i.e. the problem state supplies all the inputs needed by the algorithm. To satisfy this requirement new information is added to the problem state - the *alternative enumeration state* which keeps track of process alternatives and the *structure generation state* which helps enforce the cycle of operations and provides structural information required by the operations. The *stream state* also gains some new information which is in turn inherited by the process stream front.

The cycle of operations

Figure 3.6 shows the cycle of operations responsible for the generation of the structural alternatives. Each operation draws its input set from the current process stream front, tests for failure and adds the appropriate units to the flowsheet. Internal blending occurs before the central transformation operation in the cycle to provide the opportunity for feed stream blending. Target acceptance branching (not shown in diagram) is sandwiched between each of the operations, though it only affects structure generation when targets are satisfied.

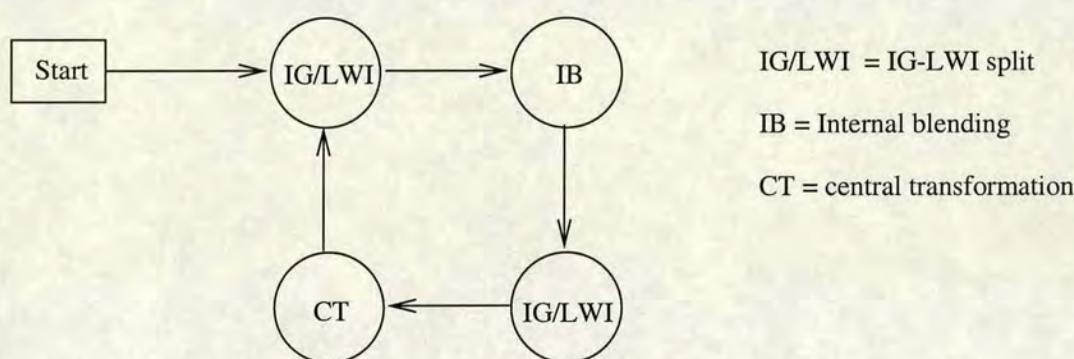


Figure 3.6: The cycle of operations

Alternative enumeration state

Within each operation there are different types of alternatives. So far only the structural alternatives have been covered. The design alternatives, which can only be created after a structure has been chosen, will be described in section 3.4. Each alternative must be fixed within the current operation before moving onto the next operation. Taking the CT operation as an example, the CT unit type must first be set, followed by the design of the unit. Only after a successful unit design can the next operation in the cycle (target acceptance branching) be executed. Within an operation the output from the current alternative forms the input to the next alternative. In order to keep track of the alternatives the *alternative enumeration state* is added to the problem state. This consists of:

- Flags to indicate the first and last alternative. The former is used to initialise the information for alternative enumeration. The latter indicates that enumeration is complete.
- The number of alternatives which must be explored and the number of alternatives which have been explored.

Structure generation state

The information required to drive the structure generation algorithm is grouped together in the *structure generation state* which forms another component of the problem state. It includes:

- The most recently designed CT unit, used primarily to enforce the mixing restriction in internal blending.
- The CT unit type selected for inclusion in the process.
- A count of each CT unit type used in synthesis so far. This information is needed if a limit is imposed on the number of instances of CT units within a flowsheet.
- Information on the current structure generation operation - its position in the cycle of operations, the operation type and flags indicating if the operation is just starting or coming to an end. If the former is true the input set for the operation must be constructed. If the latter is true, then synthesis must progress to the next operation in the cycle.
- The inputs for the current alternative type within a given operation.

New stream state information

The stream state gains the *stream type* information, i.e. newstream, immediate gratification or loiter-with-intent needed by the operations to select the correct input streams from the current process stream front. Each LWI stream knows which stream is its

IG sibling so that it can determine if it has been consumed or not. Finally, the *target satisfaction state* of a stream is a logical variable which is set to true when a stream has satisfied a target and therefore becomes an input to the target acceptance branching operation.

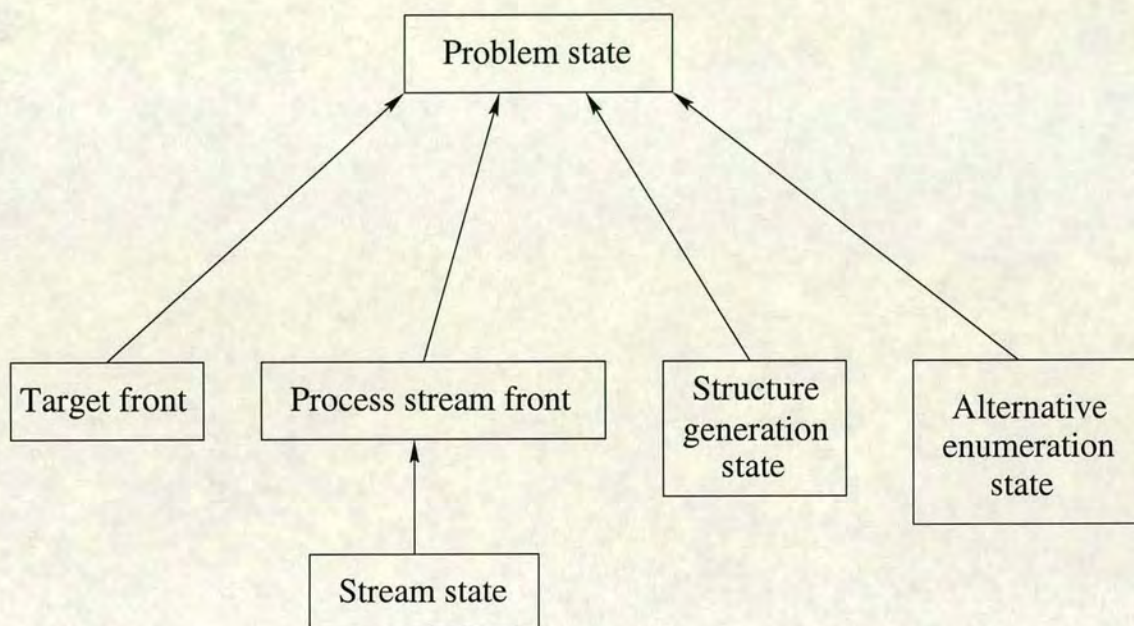


Figure 3.7: Adding more information to the problem state

3.4 Unit design space

The structural search space is defined by the cycle of structure generation operations and the alternatives embedded within each operation. The unit design search space now needs to be defined to create a combined synthesis search space which hopefully includes any and all solution flowsheets of interest. The means of setting up the unit design alternatives is first described followed by a note on how the design information is to be made available by the user.



3.4.1 Unit design alternatives

The requirements for the assignment of unit design variables are:

- An implicit enumeration approach has been used for the generation of the structural alternatives. This must be maintained for the generation of unit design alternatives.
- The method for assigning unit variables must remain true to the goals of early design, i.e. be capable of returning a set of good solutions.

These requirements are met through discretisation of the unit design variables. Each design variable has an user-defined upper and lower bound. The set of possible values it may take is defined by discretisation across that range (see figure 3.8). The result is that design variable values are now in the form of a set of discrete alternatives. Given the design variable bounds and the number of discretisation levels, the number of design alternatives for a given unit can be determined. Specifying the unit feed set and the discretisation level and distribution for each unit design variable then fixes the unit design. Thus unit design alternatives can be determined at the point of addition of a unit to a flowsheet. This fits very smoothly into the synthesis operations as another set of discrete alternatives to be enumerated.

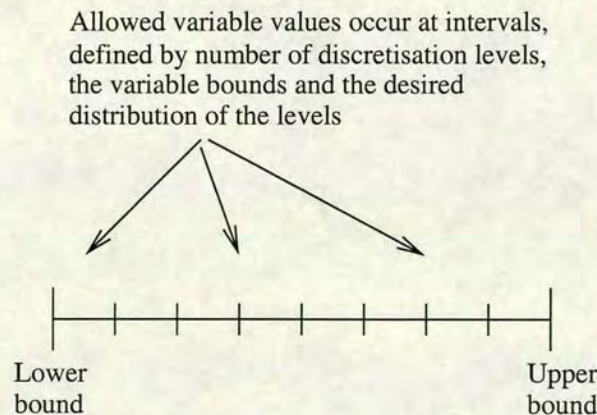


Figure 3.8: Discretisation of a unit variable across its range

Design variable discretisation is a simplification which results in a reduction of the solution space. Therefore, a search performed on this space cannot locate all solutions to a given problem since all continuous space is not being considered. However, there exist far less structural alternatives than continuous ones and covering a spread of continuous space by discretisation will locate many of these. The finer the discretisation, the greater the chance of finding the global optimum but the larger the search which must be undertaken.

3.4.2 Providing the unit design information

Unit designs require information about the process unit itself as well as the means for its solution once the design variable values have been specified. The former consists of the unit design variable set, calculated variable set and its input-output structure, i.e. the information of interest to the process flowsheet. The latter requires the use of a design procedure which will solve a fully specified unit for its outputs. Both these requirements are met by a user-provided unit model for each type of unit made available to synthesis. This allows the unit design variable discretisation parameters to be specified as well as supplying the other relevant unit information and a design procedure. More detail on the unit models, their structure and integration into synthesis is available in Appendix A.

3.5 Searching for solutions

The search space is now fully defined. It consists only of discrete decision variables and can therefore be cast in the form of a tree. This section describes the search tree created, the choice of search method for the MCP synthesis problem and the search algorithm which arises from this.

3.5.1 The synthesis tree

The synthesis tree is composed of problem states which are transformed to new problem states by operators *whose appearance and behaviour is controlled by the cycle of structure generation operations (section 3.3.4)*. The problem state is comprised of the target front, the process stream front, the alternative enumeration state and the structure generation state, i.e. all the inputs required to describe the synthesis problem, keep track of alternatives and drive the structure generation algorithm. The operators correspond to the alternative enumeration portions of each structure generation operation, eg. for the central transformation operation, they are CT unit type selection and CT unit design. Each alternative generates a different output for the same input resulting in a tree which is to be searched for solutions. The failure cases outlined for the structure generation operations together with unit design failures result in operators which have no outputs. These cases form dead ends in the search tree. Figure 3.9 illustrates a section of a synthesis tree.

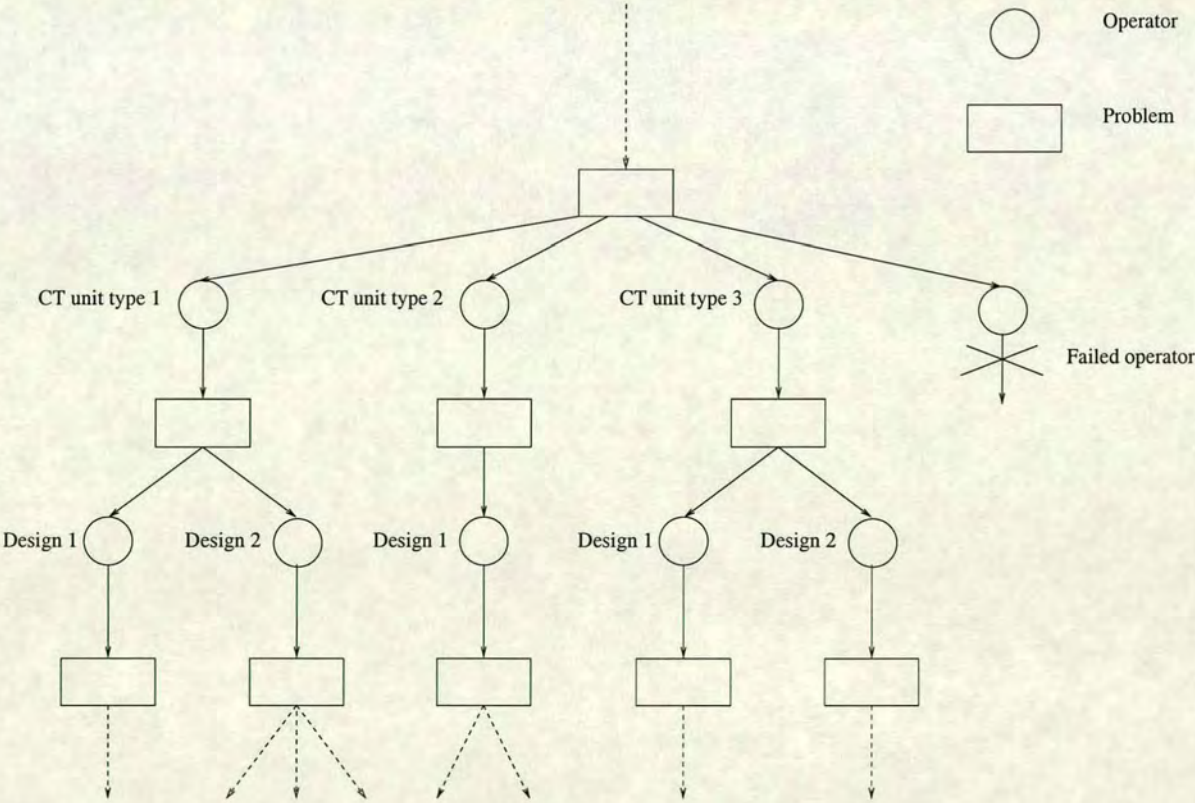


Figure 3.9: Synthesis tree

3.5.2 Selection of search method

This tree must be searched in a manner consistent with the aims of MCP design and the form of the tree itself. Experience of processes involving nonsharp separation, stream splitting and blending is very limited. Heuristics and evolutionary rules rely on such experience. Hence synthesis methods based primarily on these are unlikely to be useful. Therefore, an algorithmic technique is of interest because it is useful in the development or design of new processes. Also, an algorithmic approach is more amenable for use in the automated generation of processes which is desirable in the early stages of design. Furthermore, it is essential to have the ability to generate a set of the best solutions to a given problem. The cost optimal solution is unlikely to be optimal in terms of safety, operability etc. and the designer should be provided with several flowsheets on which to base judgemental decisions, and indeed recost using more detailed procedures.

A deterministic tree search is most suitable as it is both algorithmic and capable of returning a set of solutions. However, the correct tree search must be chosen. Tree search procedures are of two kinds - blind procedure and heuristically informed procedures. The latter are best applied to search spaces where a good measure of distance from the solution or goal exists. Since such a measure has not been established for MCP problems blind procedures are the only possibility. These consist of:

- A *depth-first search*. This is a good idea when you are confident that all partial paths either reach dead ends or become complete paths after a reasonable number of steps. In contrast, a depth-first search is a bad idea if there are long paths, even infinitely long paths, that neither reach dead ends nor become complete paths.
- The *breadth-first search* works even in trees that are infinitely deep or effectively infinitely deep. On the other hand, breadth-first search is wasteful when all paths lead to a solution at more or less the same depth.
- A *nondeterministic search* is used when one is unsure as to whether a depth-first search or a breadth-first search would be better, i.e. when it is not known

whether the search tree has either a large branching factor or long, useless paths. A middle-ground approach is employed by the nondeterministic search such that paths are explored from random starting points, ensuring that one cannot get stuck chasing either too many branches or too many levels.

These search methods only exhibit different behaviours when parts of the tree are pruned. This involves the use of bounding - if the cost of the current partial path is such that possible solutions arising from that path can be no better than some complete solution already obtained, then it is not necessary to develop the alternatives available at that point, i.e. that section of the synthesis tree is not searched for solutions. As a result, pruning can reduce the search space and hence the solution time - though the extent of this reduction varies from problem to problem.

To reduce the search space for the MCP synthesis problem, it was decided to control the depth of the tree by introducing a maximum number of allowed instances of each available CT unit in a flowsheet. As a result, all partial paths within the synthesis tree either reach a dead end or become a solution before the CT unit limit is exceeded. A depth-first search is the right search for such a tree.

3.5.3 Search algorithm

The search algorithm employs a depth-first search to locate the best solutions within the synthesis tree. The size of the solution set is specified by the user and the solutions are ranked in order of increasing cost. The user can elect to carry out a complete search of the tree or decide to introduce pruning. When the latter is switched on this activates the storage of solution costs as global bounds, i.e. this information is accessible at all times. However, no pruning of the tree occurs until the set of global bounds has reached its specified size, i.e. partial path costs are not compared against the worst cost solution (the *worst incumbent*) until after this point. Figure 3.10 shows the algorithm as implemented for this work.

This algorithm is recursive and will traverse the tree downwards always choosing the

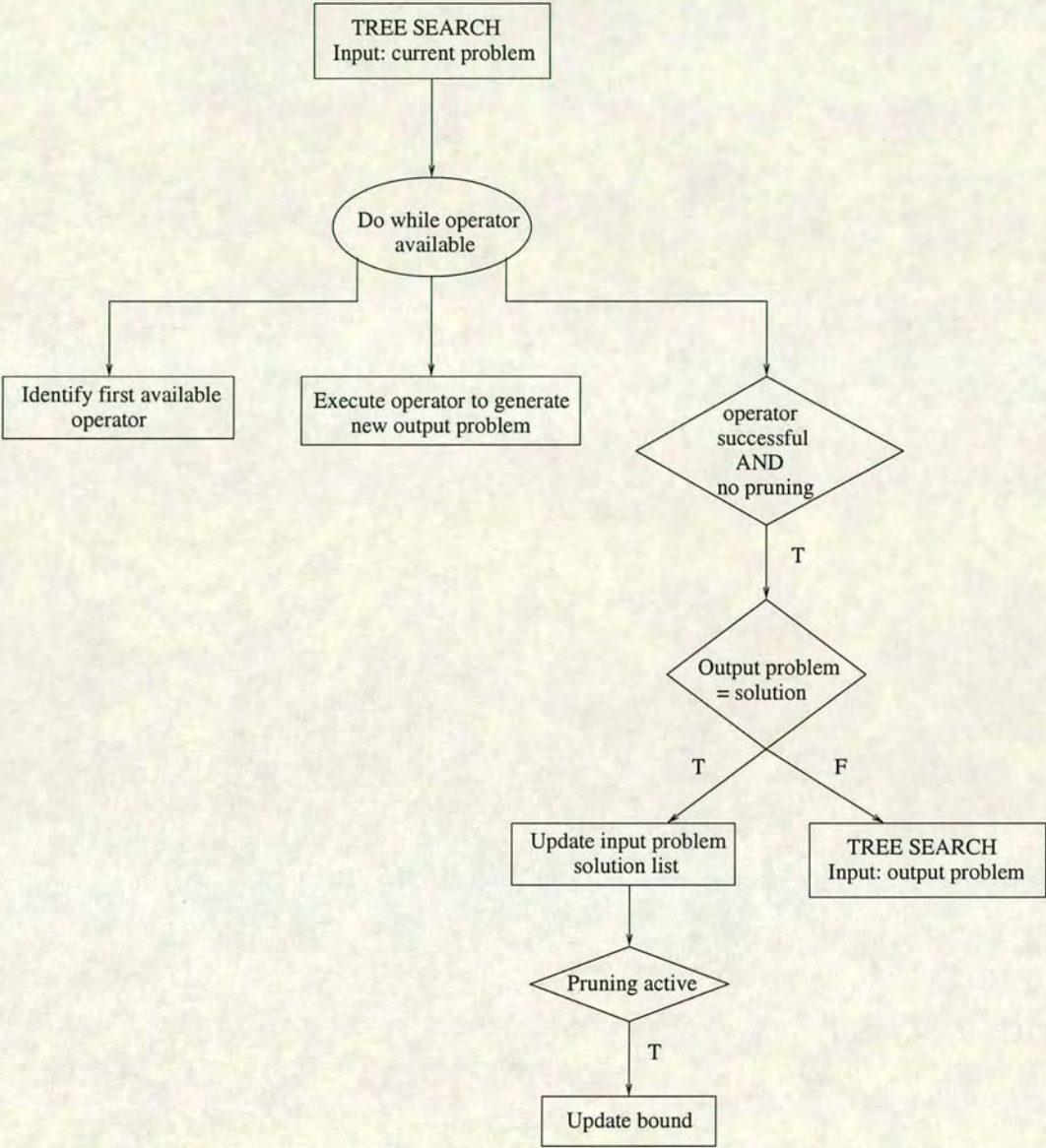


Figure 3.10: Depth-first tree search algorithm

first available operator (process alternative) to generate a new output problem. When a dead end is reached through failure of an operator, eg. exceeding the CT limit, unit design failure, the next operator available to the input problem is attempted. Pruning, if activated by the user, a full solution set and an output cost exceeding the worst incumbent, effectively imposes a dead end by not exploring the alternatives available to the output problem of a successful operator. Instead, the search returns to the next operator available to the input problem.

Streams which have been accepted as targets are removed from the process stream front. An output problem where all targets have been satisfied and the PSF is empty is a solution and is not operated on. It is added to the input problem solution set if that set has not yet reached its specified size or if its cost is less than the worst solution in a full set. For the case where pruning is active the solution is added to the set of global solution bounds subject the same conditions. When all the alternatives available to the input problem have been explored the search retraces its path to the first problem with an unexplored alternative. The act is referred to as *backtracking*. When there are no unexplored alternatives the search has been completed. The structuring of the synthesis problem information, the data types used to describe the synthesis tree and the tree search algorithm code are available in Appendix A.

3.6 Removal of redundancies

Redundancies are repetitions of process structures and are undesirable because they needlessly increase the search space and solution times for synthesis problems. Neither the design search space nor the structural alternative search space contain redundancies by themselves. However, the combined spaces contain symmetry in the assignment of split fraction values which leads to repeats of particular structures. This section describes the redundancies identified and the procedures used for their removal.

Internal blending redundancies

New intensive stream states (created by the last CT unit added to the process) must have the opportunity to be involved in the internal blending operation. Existing intensive stream states can bypass this operation by being *completely* LWI streams, resulting in no blending. However, if new intensive stream states are also allowed to *completely* bypass this operation, there now exist two cases which result in no blending as figure 3.11 illustrates. There are more redundant structures based on case 2 (eg. only one of the newstreams bypasses blending) and these must also be ruled out.

Redundancies effectively form failure cases for the operation involved as procedures for removing them involve testing for the redundancy, followed by failure of the operation if the test is positive. For internal blending the procedure used to remove redundancies is as follows:

1. Look at the LWI streams in the process stream front entering the operation.
2. If any of the LWI streams created by the last CT unit added have zero flow IG siblings then the operation fails.

This procedure ensures that the outputs from CT units cannot completely bypass the internal blending operation.

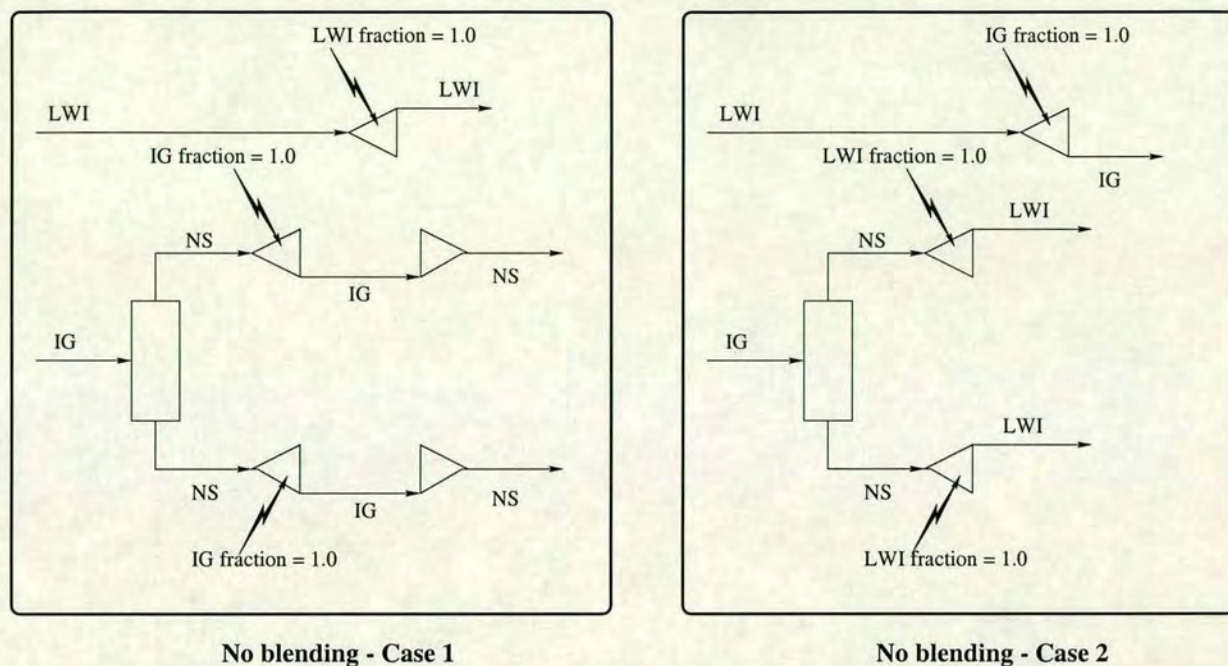


Figure 3.11: Internal blending redundancies

Target acceptance branching redundancies

It is important to carry out target acceptance branching only once on a given stream, i.e. to not repeatedly branch on the same product stream. This is avoided as follows:

1. Every stream that has satisfied a target has a logical flag attached to it indicating that it has done so - though it may not have been accepted as a target and therefore remains in the process stream front.
2. Each output from the IG-LWI split and internal blending operations is checked to see if it has the same state as one of the input streams. If so, the value of the target satisfaction flag is copied from the appropriate input stream to the output stream.
3. Streams which remain in the process stream front but have satisfied a target are not retested for target satisfaction and are not entered into the input set for target acceptance branching.

This procedure ensures that a stream is selected as an input to the TAB operation only once.

Central transformation redundancies

The IG-LWI split before the central transformation operation can lead to symmetrical structural alternatives which have the same input and output problem states. Only one such alternative is needed, the remainder are redundant. In examining these redundancies it is assumed that the internal blending operation is always bypassed (as in Case 1 of no blending described earlier). As a result, no blenders appear in the example structures given. The two types of redundancies encountered are shown in figures 3.12 and 3.13.

The first type of redundancy shows a stream undergoing processing by a parallel structure - it is split into two nonzero fractions with each fraction destined for a different CT unit. However, the symmetry implicit in binary split fraction values can result in two different alternatives which contain the same CT units and the same output process stream fronts. The only difference between the structures is the order with which each stream fraction is processed by a CT unit.

The next redundancy occurs when there is more than one stream in the current process stream front and when values of 0.0 and 1.0 are both available for the IG-LWI split fraction. Referring to figure 3.13, the split fraction alternatives can make the same stream state available to the same CT unit more than once. The resulting structures are symmetrical, contain the same CT units and have the same output process stream fronts. Again, the difference between the structural alternatives is the order in which each of the input streams is processed by a CT unit. Both types of redundancies are manifestations of the same problem: *the structure generation algorithm combined with the split fraction symmetry can lead to alternative structures which differ only in the order in which streams are processed by CT units.*

The procedure for removing the CT redundancies relies on the fact that, regardless

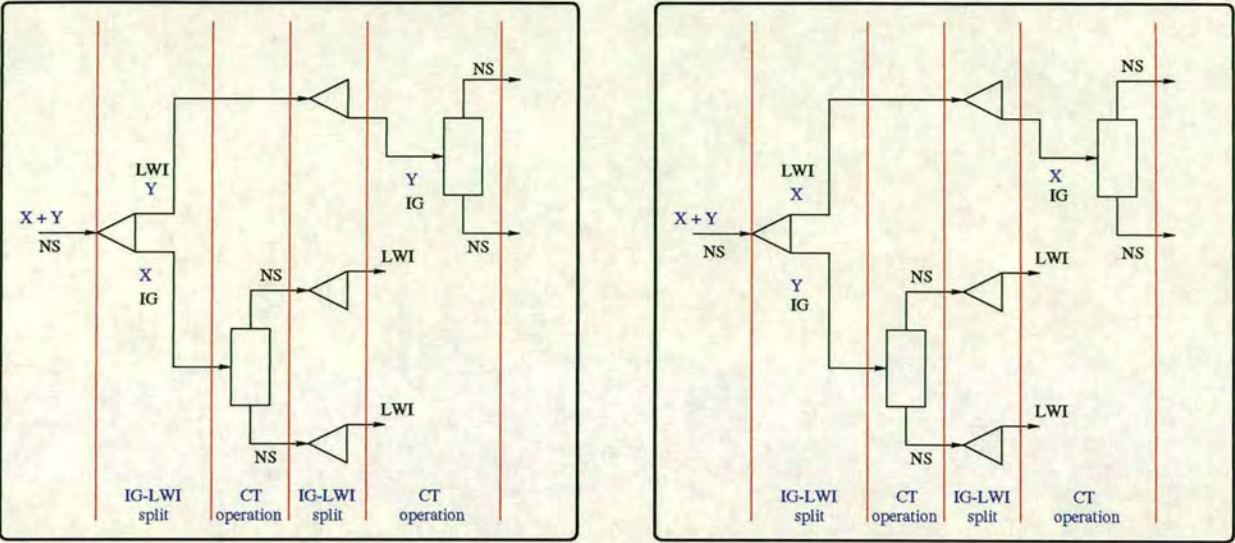


Figure 3.12: CT redundancy - first type

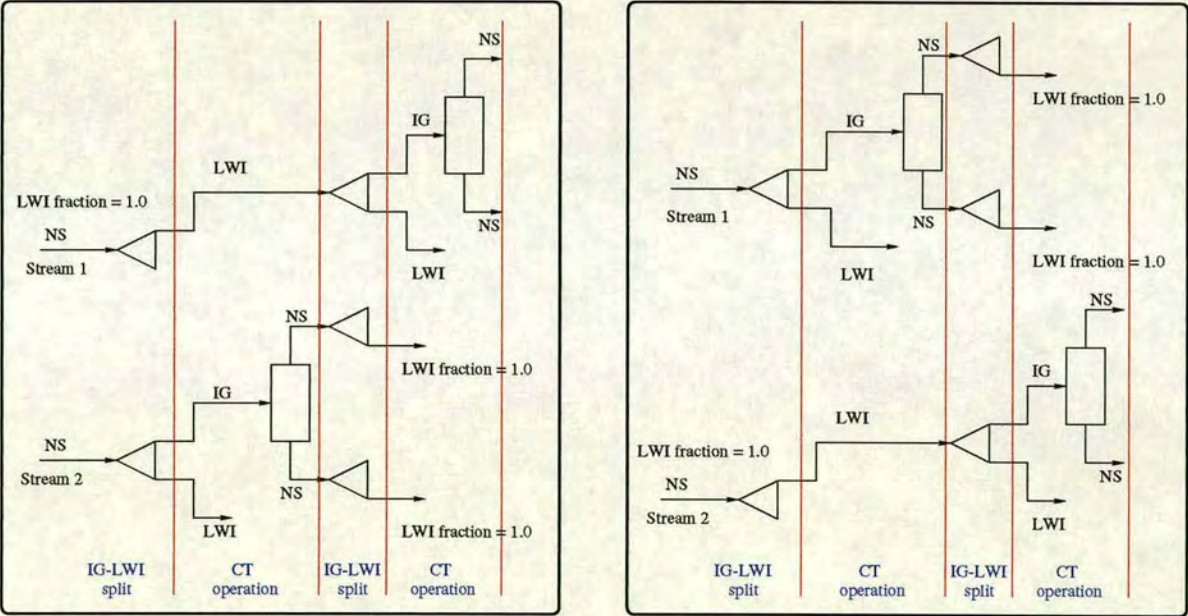


Figure 3.13: CT redundancy - second type

of the search method used, synthesis alternatives occur in a specific order. Once one alternative involving CT units has been fixed this can be used to eliminate its redundant siblings. This is achieved as follows:

- Once a given stream has been chosen as the input to a given CT unit this information is propagated backwards to the input of the IG-LWI splitter which set the

destination of the stream. The information includes the type of CT unit involved, the fraction of the splitter input assigned to the CT unit and a logical flag indicating that this intensive stream state has been considered as an input to a CT unit.

- When the splitter input is encountered again, the CT alternative information is propagated to the splitter outputs. If these streams are subsequently considered as CT unit inputs they are first checked to ensure that the same stream fraction and CT unit type are not involved. If so the stream is processed, otherwise the CT unit addition is failed.

In general, the CT usage information is added to every stream encountered in synthesis and therefore forms a new component of the synthesis stream state. IG and LWI pairs take on the CT usage information attached to the inputs to the splitters which created them. Newstreams are initialised to indicate that they have not yet been considered as a CT unit input.

3.7 Completeness of the structure generation method

In advance of attempting synthesis problems it is necessary to know the range and diversity of structures which the synthesis method makes available. It is also worth assessing whether obvious or important structures are being omitted. The structure generation method is examined with these aims in mind and is approached by first looking at the possible CT unit configurations followed by an analysis of the available blending structures.

3.7.1 CT unit structural alternatives

The blending operation does not affect the appearance of CT units in the process. It makes new stream states available and thus provides more possible inputs to CT units. However, the creation of the CT unit alternatives for a given process stream front is

not influenced by streams' blending histories. It is the IG-LWI split which controls the stream destinations and this operation together with the CT operation sets up the CT unit configurations. Therefore, without losing generality, blending can be omitted when examining the CT unit structural alternatives generated by the synthesis method.

All CT unit structural alternatives are generated if all possible feed stream sets are made available to the current CT unit and if all available CT unit types are attempted for each feed set. The minimum necessary requirement to meet the first condition is that the IG-LWI split before the CT operation *always* has zero and a nonzero value as possible IG split fraction values. These values meet the need for a discrete decision before the CT operation - that of providing IG sets of different sizes and combinations of each intensive stream state available. The second condition is satisfied by making CT unit type available in turn for each feed set to the CT operation - this is one of the structural alternatives which is built into the this operation.

Where split fractions of zero and one are used only sequential processes, such as the direct and indirect sequences commonly used in pure component product processes, are possible (figure 3.14). For cases where a split fraction between zero and one is made available, parallel structures where the same intensive stream state is processed by many different paths (figure 3.15) can be obtained.

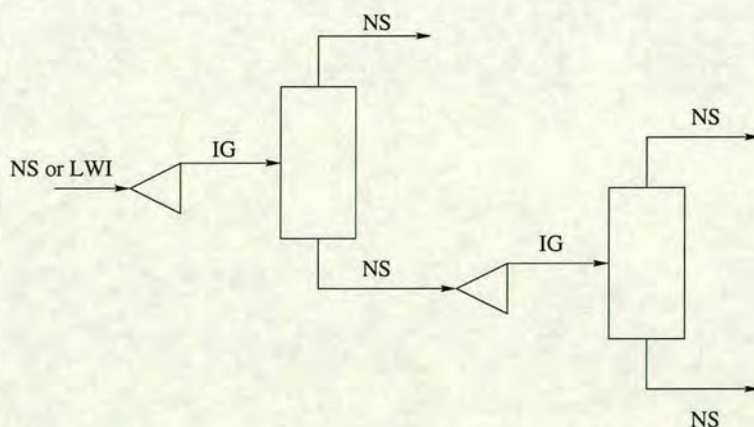


Figure 3.14: An direct separation sequence

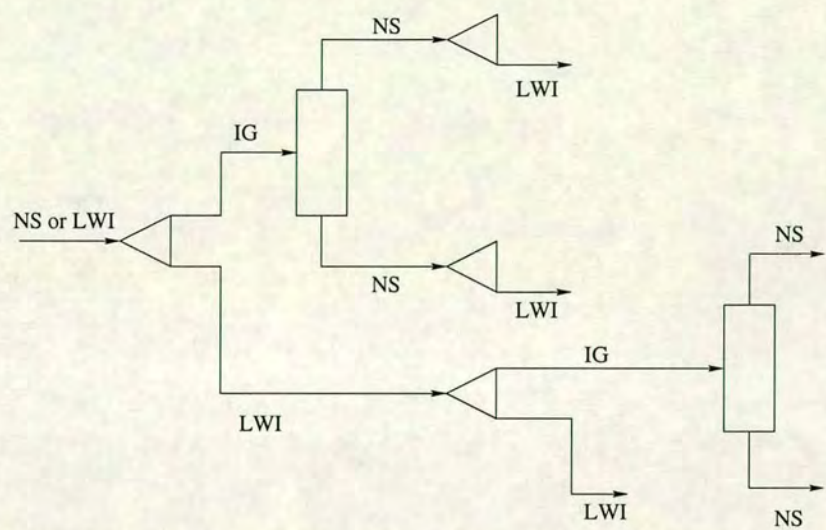


Figure 3.15: Creating a parallel structure

3.7.2 Blending structural alternatives

The available blending alternatives are examined for three different cases - feed stream blending, blending in the separation section, and blending before the products. The block diagram (figure 3.16) of an MCP process shows these different areas.

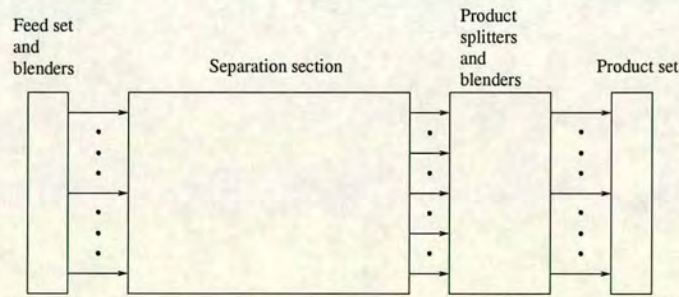


Figure 3.16: Block diagram of separation process

Feed stream blending

This is handled by the synthesis method in a limited manner. Each feed stream is given a fictitious CT parent, one of which is set to be the current CT unit. Thus, one blender is declared for which one of the feed IG stream states is the input. The remaining feed streams are distributed to that blender. Figure 3.17 illustrates the resulting structure for four feed streams. Such a structure is sufficient to cover the all blending alternatives

for two feed streams, i.e. the structure can access all stream states available at that point. However, above that number this is not the case. In the four feed example provided, the blending of feed A with feed B and the blending of feed C with feed D is not possible. Many other such alternatives are not available. It would seem that for m feed streams that $m - 1$ blenders are needed to cover all these alternatives - though this has not yet been implemented.

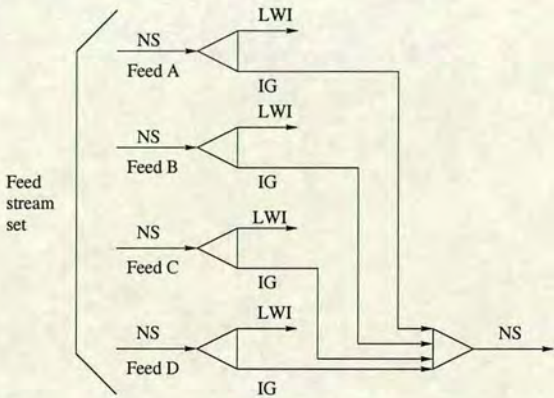


Figure 3.17: Blending of four feed streams

Blending in the separation section

This is the blending which the internal blending operation was designed for. Its purpose is to access new stream states bounded by the feed streams and by the CT unit output streams. Blenders are associated with these boundary states, i.e. the appearance of blenders corresponds to the appearance of the CT unit output streams. Internal blending looks for the IG streams created by the current CT unit and makes each a feed to a blender. All other IG streams in the process stream front at that time are distributed between those blenders. The restriction against mixing the outputs of the same CT unit prevents the wasting of separation work, or compression work, or whatever energy has been spent on establishing the CT unit output states. Internal blending sets up a large number of blending alternatives. It can mix a stream created by any earlier CT unit in a process with the output from the current CT unit.

Again, as with the CT alternatives, the IG-LWI split is largely responsible for making

the different blending structures available. Outputs from earlier CT units blend with those from the current CT unit if they have a nonzero flow IG fraction on entry to the internal blending operation. If they have a zero flow IG flow they are not involved in blending. The minimum set of zero and a nonzero value for IG split fractions reappears as the means of making the different structures available.

Product blending

Product blending is envisaged as occurring outside the separation section of an MCP process. Each product has a blender associated with it and the outputs from the separation section will have these as their final destinations. Internal blending only associates blenders with the CT unit output streams. This can provide some product blending after the last CT unit. However, unless the number of output streams from the last CT unit added equals the number of products, it is not possible to guarantee that each stream state in the process can have a product blender as a possible destination.

There appear to be two approaches to the inclusion of product blending alternatives. Firstly, determine which (if any) of the products are bounded by stream states in the current process stream front, i.e. which products are mixtures of some fraction of the streams in the process stream front. A blender could then be introduced for each bounded product and the split fractions required to obtain them determined. The second approach is to use the stream bypass operation which occurs in the literature methods. When a new stream state is created a fraction of it can be bypassed to the product blenders. This has the result of altering the product specifications such that they now correspond to the second blender input which the process stream input and original product stream output must give rise to (figure 3.18). This approach would lead to a very large number of alternatives unless targetted as in the literature methods. Both of these possible approaches may run into difficulty when dealing with the region of stream states defined by effect products, especially so if the effect property is nonlinear.

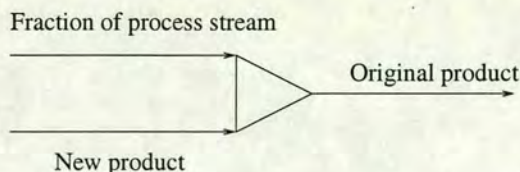


Figure 3.18: Product bypass alters product specifications

3.8 Summary and conclusions

The synthesis method described takes a very basic definition of unit function (its contribution to stream state diversity) and uses this as a foundation for an algorithm which generates structural alternatives incorporating separators, splitters and blenders. The unit design variable alternatives are created through design variable discretisation which results in a search space consisting only of discrete decisions thus permitting it to be cast in the form of a tree. A depth-first search algorithm with pruning is used to explore this tree for a user-specified number of solutions.

The method meets the requirements for an early design stage synthesis tool as outlined at the end of the last chapter. It can handle effect product specifications through the use of a simple problem state definition which does not constrain target properties. The structural alternatives are generated systematically with opportunities for stream splitting, nonsharp separation and blending. The search method used is algorithmic and can return a set of solutions, making it suitable for the design of new processes. The method makes a much richer set of process alternatives available than most existing methods and does so without placing a large workload on the designer. Once the feed set, product specifications and process unit design information are supplied, the method simultaneously generates and searches through the process alternatives.

Chapter 4

MCP Synthesis Method Results

4.1 Introduction

The MCP synthesis method is applied to three multicomponent product problems. Two literature problems requiring nonsharp separation of a ternary hydrocarbon feed are presented for the purpose of demonstrating that the synthesis method is capable of generating a richer set of solution structures than the current literature methods. Finally, a case study of industrial scale and interest is carried out. This problem has twelve feed components, an effect specification on one of the products and requires the use of nonsharp separators. Such a problem could not be attempted using the approaches described in the literature.

4.2 Unit models

The unit models are responsible for solving the process units which are added to flow-sheets during synthesis. For the given unit type, its feed stream and the values of its design variables, these models supply the unit design, its output stream states and costs. Splitters and blenders possess unit models but their solution is straightforward and they are assumed to have zero cost. The units involved in central transformation operation are of greater interest and importance. For this work, the CT units used are:

- Flash vessels - these are the nonsharp separators which are made available for each of the problems.
- Compressors are required to meet a product delivery pressure specification for the industrial case study.
- Distillation columns are used for sharp and semi-sharp separations.

The thermodynamics used for the MCP problems are first outlined and are followed by a description of each of the unit models made available to synthesis.

4.2.1 Thermodynamic and physical properties

Each of the MCP problems presented have hydrocarbon feeds. Ideal thermodynamic properties can be assumed and component properties are estimated using the Kistiakowsky method [19] for hydrocarbon mixtures. Procedures for the estimation of physical properties (density, viscosity etc.) are obtained from Perry's Chemical Engineers' Handbook [31]. The flash solutions obtained using the ideal thermodynamics compare well with ASPEN PLUS simulations using Redlich-Kwong-Soave thermodynamics.

4.2.2 Flash model

The flash unit is a single-input, two-output separator and possesses two design degrees of freedom. The flash temperature and pressure are used as the design variables as these provide the most straightforward route to solution. Within the model, the assignment of design variable values is carefully controlled so as to make the best use of the alternatives considered.

The flash pressure is set first and the values used are distributed logarithmically between the upper and lower bounds provided by the user. Such a distribution is suitable since vapour-liquid equilibrium separations are not very sensitive to pressure and thus require larger pressure changes to obtain significant changes in unit outputs. Having set the value for the pressure, the model checks if this value is valid. Two cases are ruled out:

1. Operation of the flash vessel at a higher pressure than its feed stream. Compressing a vapour feed is a far more costly means of operating a flash in the feasible two-phase region than cooling. In the case of a liquid, pumping moves the feed state further away from the feasible operating region, increasing the amount of heating required to perform the separation.
2. The second set of pressures not considered are those which are lower than the pressure of a vapour feed. These values move the feed state further away from the two-phase region which the flash vessel must operate in, increasing the amount of cooling required to perform the separation.

The flash temperature upper and lower bounds are adapted to the feed composition and pressure, i.e. the dew and bubble point temperatures of the flash feed at the flash pressure become the upper and lower operating bounds respectively. In this way no design temperature value is outside the feasible range. Flash temperature values are distributed uniformly between the upper and lower bounds. Since the bounds are never very wide and the sensitivity of the separation to temperature changes is significant, picking points evenly across the range is a reasonable strategy.

An important alternative to consider is the flashing of a feed stream at its own temperature and pressure as this does not incur any utility costs. For the case of pressure, the discrete level which results in a pressure setting nearest to the feed value is located. If this is the current level then the flash pressure is set to the feed pressure. Otherwise, the value dictated by the discretisation level and distribution is used. In order to consider operating at the feed temperature, the feed stream is checked to determine if it is a two-phase mixture at the flash pressure and feed temperature. If so, the flash temperature is set in the same manner as the pressure. Otherwise its value is set purely by discretisation.

The solution of the flash is rigorous (insofar as chosen thermodynamics permit). The design fixes the output streams for the given feed and values of the design variables. Mechanical design and costing are then carried out. A shortcut design of the heat exchangers associated with a flash vessel is performed. The utilities available are cooling

water at 288K and steam at 14 bar. The capital and utility costs of the exchangers are added to the total vessel cost. Unfortunately, as the procedures used for the mechanical design and the costings are proprietary, they cannot be described in more detail. However, they provide good cost estimates and are reasonably up-to-date.

4.2.3 Compressor units

The industrial case study problem calls for the use of compressor units. Their function in the flowsheet is to meet a gas product pressure specification. The compressor unit is restricted to vapour feeds and the design is failed if a feed of any other phase is supplied. The design variable is the outlet pressure and this is distributed logarithmically between its lower and upper bounds. There are two types of compressors available - for outlet to inlet pressure ratios above 5.0 or power required above 30MW (industry source), reciprocating compressors are chosen over their cheaper centrifugal counterparts.

For the compressor unit design, the assumption of ideal thermodynamics results in a constant c_p/c_v ratio for the compressor work formula (equation 4.1).

$$Work = RFT_1 \left(\frac{\gamma}{\gamma - 1} \right) \left[\left(\frac{P_2}{P_1} \right)^{(\gamma-1/\gamma)} - 1 \right] \quad (4.1)$$

where R is the ideal gas constant (kJ/kmolK), F is feed stream flow, c_p is the vapour heat capacity at constant pressure (kJ/kmolK), c_v is the vapour heat capacity at constant volume (kJ/kmolK), γ is the c_p/c_v ratio, T_1 is the feed stream temperature (K), P_1 is the feed stream pressure (bar) and P_2 is the compressor outlet pressure (bar). The c_p/c_v ratio for compressor design must be supplied by the user and is assumed to be constant for all designs. This assumption is valid if the composition of the inlet to each compressor is approximately the same.

Rough intercooling is also performed: if the temperature of the inlet stream to a compressor exceeds the dewpoint temperature of the stream at the inlet pressure then that stream is cooled to its dewpoint temperature. The use of intercooling introduces a

design trade-off (number of compressor stages versus temperature rise) which synthesis should be aware of. The capital and utility costs used are from the same source as those for the flash model.

4.2.4 Distillation column model

The distillation column model uses the shortcut method described by Rathore et. al [33] for the design and costing of the column separations. It is applicable to sharp and semi-sharp separations and uses as its design variables the column pressure, the light key recovery in the tops and the heavy key recovery in the bottoms. The final design variable is the separation breakpoint. This is an integer variable, the value of which sets the light and heavy key for a separation. The separation feed components are ordered by volatility and the breakpoint corresponds to a separation between adjacent components in the ordered list. The bounds on its value are 1 and $nc - 1$, where nc is the number of components in the feed. Use of the breakpoint as a design variable allows a single model to provide the full range of sharp and semi-sharp separations for adjacent key components.

Knowing the feed stream and the design variable values, the shortcut method of Rathore et al. [33] uses the Fenske equation to determine the minimum number of stages and the Underwood equation for the minimum reflux ratio. According to Andrecovich and Westerberg [3] the economically optimal reflux ratio usually lies in the range of 1.1-1.25 times the minimum. A value of 1.2 is used for this work. Finally, the Gilliland correlation determines the actual number of stages. Having fixed the reflux ratio and number of stages the correlations supplied by Rathore et. al are used for the column mechanical design and capital cost estimate.

In order that the column costs are comparable to the flash vessel and compressor costs, the Marshall and Swift indices [25, 26] are used to update the capital costs from 1968 to 1995, the latter being the year to which the flash vessel cost correlations apply and the former the column cost correlations. Furthermore, the column costs are converted from US dollars to UK pounds (used by the flash correlations) using the average 1995

exchange rate of \$1.578 to the UK pound (source: US Federal Reserve web site).

The same heat exchanger unit design and capital costs as used for the flash vessel design are applied to the column heat exchangers. Finally, the same utilities, cooling water at 288K and steam at 14 bar, are used throughout. Their costs form the distillation column operating costs.

4.3 Literature problems

All the literature methods encountered use distillation columns in the solutions to MCP separation problems. These columns perform sharp and semi-sharp separations to produce nonsharp product sets. In cases where the products involved do not contain all the feed components such column separations are justified, i.e. a complete separation is needed between some of the feed components. However, for cases where all or some of the products contain all feed components, solutions could be obtained through the use of nonsharp separators. The literature approaches produce such product sets through the use of the bypassing operation (sections 2.2 and 3.7) coupled with sharp and semi-sharp separations. This produces a trade-off between the amount of material passing through distillation columns and the sharpness of separation. However, the lower bound for the latter is always in the region of 80%-85% for the value of the keys recoveries in the outputs. Nonkeys are never permitted to distribute between the outputs.

The MCP synthesis method presented in this thesis can produce bypassing structures if the number of products equals the number of outputs from the type of separator chosen (section 3.7.2). For the following problems this is the case, allowing the synthesis of flowsheets similar to the literature solution flowsheets. However, the method creates a much richer set of alternatives than most of literature approaches, most notably the capacity to include nonsharp separators. It is of interest to examine how such separations perform on the example problems encountered in the literature.

Aggarwal and Floudas [2] are one of the few authors using real components rather than hypothetical ones. Since the intention here is to use flash vessels, which require real

component data for vapour-liquid equilibrium calculations, these example problems are used. Aggarwal and Floudas employ a superstructure to formulate a mixed integer nonlinear programming problem and the solutions they obtain contain sharp and semi-sharp separations, stream splitting, blending and bypassing.

4.3.1 Problem 1

This problem requires the separation of a ternary hydrocarbon feed into two nonsharp products. The component recoveries in the products indicate that they could be obtained in a single separation, as Aggarwal and Floudas demonstrate.

Feed stream data

Component	flow (kmol/hr)
Propane	100
i-Butane	100
n-Butane	100

Table 4.1: Aggarwal and Floudas Problem 1 - Feed stream component flows

The feed temperature and pressure are not provided by Aggarwal and Floudas. Values must be assigned for the distillation column and flash vessel designs. The units use water at 288K as the cooling utility. However, operation of the separation units at atmospheric pressure will mean that this utility cannot be used as the temperatures which guarantee feasible operation are well below 288K - the boiling point of the least volatile component, n-butane, at 1 atmosphere is 272.5K. Operation at 13bar results in a boiling point of 308K for propane. This provides a good temperature driving force for the operation and design of the heat exchangers for all streams including a pure propane column output. Aggarwal and Floudas provide no operating pressure information, though they use a cooling utility at 305K which indicates that this pressure is above atmospheric. Therefore, it has been assumed in the absence of other data, that the feed temperature is at 298K (ambient temperature), its pressure is 13bar and all separations occur at that pressure.

Product specifications

Aggarwal and Floudas specify the two products in terms of exact component flows. These are not viable specifications for the MCP synthesis method since it uses discretisation to set the design variable values. As a result, it is unlikely that the exact designs will be obtained such that those component flow specifications are satisfied. Instead, the products are specified in terms of component flows with broad bounds for their satisfaction. The flowsheets which meet these product flow bounds can then be fine tuned for better satisfaction of the original product specifications

	Products (kmol/hr)	
Component	Product 1	Product 2
Propane	80	20
iso-Butane	30	70
n-Butane	20	80

Table 4.2: Product set data

Specification	Lower bound	Upper bound
Propane flow (kmol/hr)	70.0	90.0
iso-Butane flow (kmol/hr)	20.0	40.0
n-Butane flow (kmol/hr)	10.0	30.0

Table 4.3: Product 1 specifications

Specification	Lower bound	Upper bound
Propane flow (kmol/hr)	10.0	30.0
iso-Butane flow (kmol/hr)	60.0	80.0
n-Butane flow (kmol/hr)	70.0	90.0

Table 4.4: Product 2 specifications

Aggarwal and Floudas' solution

The optimal sequence obtained [2] consists of a semi-sharp separation with a feed bypass about the column to the product blenders. The authors' results show that, although the use of sharp separations results in less material flow through the separation, solution flowsheets require two columns. Semi-sharp separation allows the product set to be obtained in a single separation resulting in a large capital cost saving. The solution is shown in figure 4.1.

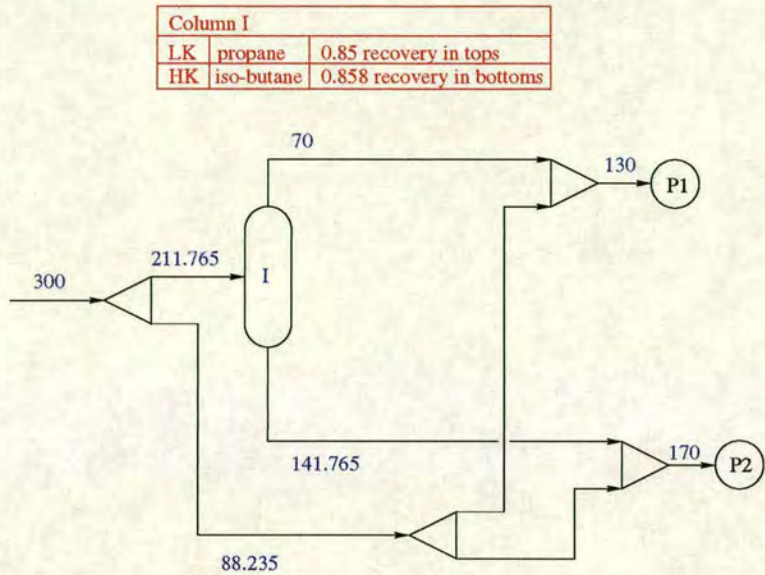


Figure 4.1: Aggarwal and Floudas' best solution for problem 1

Semi-sharp separation solutions

The aim was to demonstrate initially that the present MCP method can generate solution structures similar to the optimum returned by Aggarwal and Floudas. This particular structure also provides a good overview of the cycle of structure generation operations in action.

A single instance of the distillation column model is made available. It is permitted to try a single breakpoint resulting in propane as the light key and iso-butane as the heavy key. The keys recoveries are both set to 0.85. The column operating pressure is 13 bar. The splitters involved in the process are assigned 4 levels for the split fraction, resulting in possible values of 0.0, 0.333, 0.667, 1.0. A plant life of 1 year is specified, pruning is activated and synthesis was asked to return the 5 best solutions. Since the solution is known in advance and it is not likely to be greatly improved on only the alternatives needed to obtain it are supplied.

A single solution was returned, of the same structure that of Aggaral and Floudas with a product set close to the desired composition. The flowsheet is illustrated in figure 4.2 with the structure created by each structure generation operation highlighted. The

cost of the solution is 253,975 UK pounds with approximately equal contributions from the capital and operating costs.

The synthesis method took a large amount of time to complete the search for this solution - over 5 hours of which the search time comprises 95% and the unit design the remainder. The 4 split fraction levels within each of the 3 IG-LWI operations sets leads to an explosion in the size of the search space.

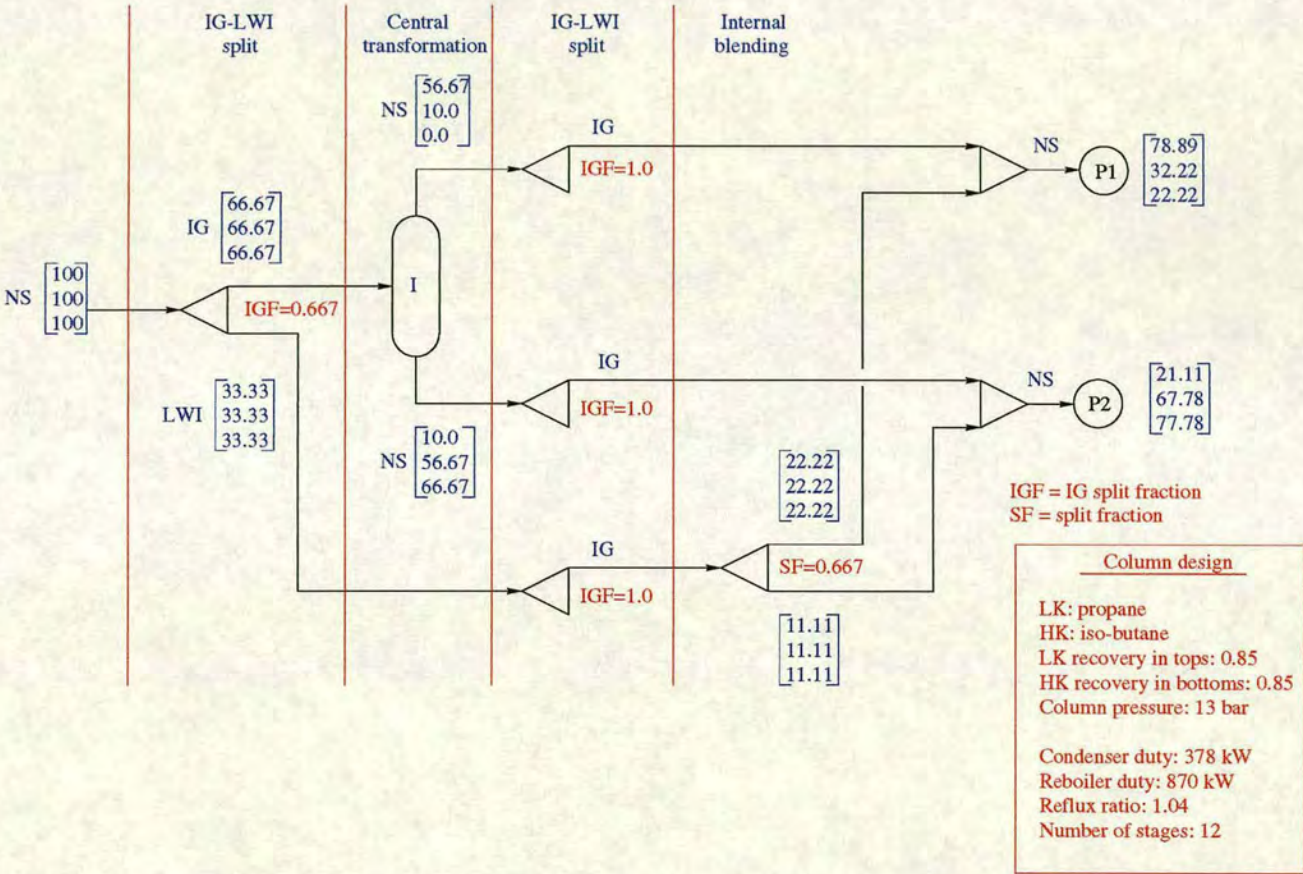


Figure 4.2: Semi-sharp solution to problem 1

Non-sharp separation solutions

Since the product set is nonsharp is makes sense to attempt to arrive at it using nonsharp separations. Up to 5 flash vessels are made available, all operating at 13 bar, i.e. the upper and lower pressure bounds are 13 bar and 1 discrete level is made

available. The flash temperature is assigned one discrete level which results in a value in the middle of its feasible range. Since 5 central transformation units will result in quite a large search space the split fraction alternatives are reduced to 2 discrete levels, resulting in values of 0.0 and 1.0.

The synthesis method ran for 25 minutes of which the design time took up 8% and the search time the remainder. Two solutions were returned both containing 5 flash vessels. The costs for each are shown in table 4.5 and the best solution returned is illustrated in figure 4.3.

Solution	Cost (UK pounds)	Relative cost
1	402,786	1.00
2	411,975	1.02

Table 4.5: Nonsharp solution costs for problem 1

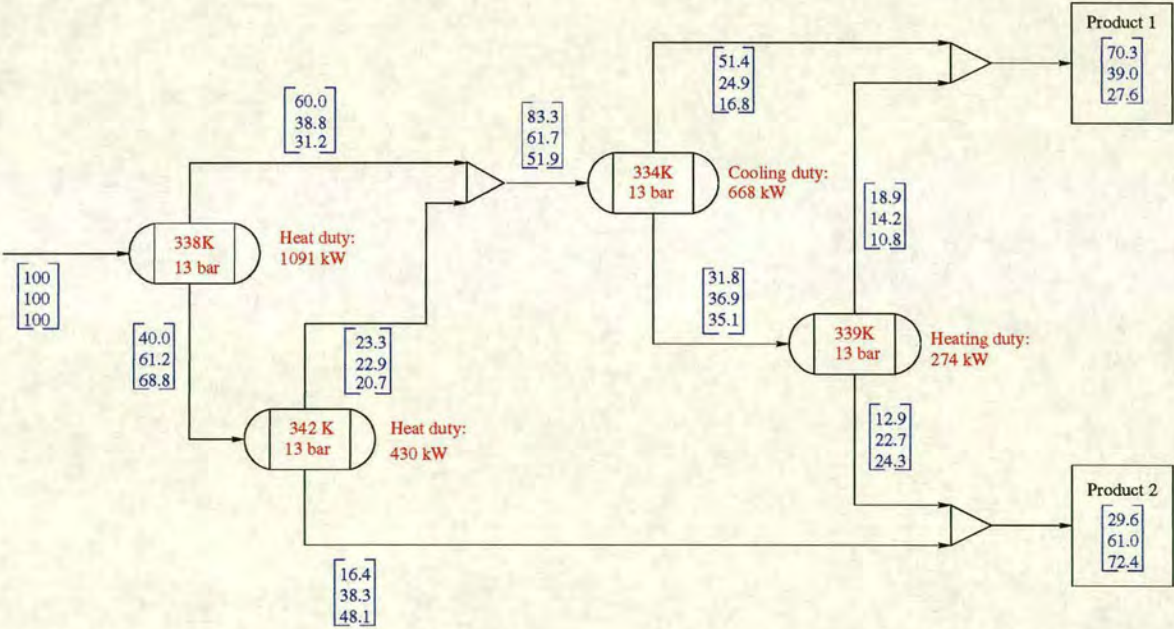


Figure 4.3: Best nonsharp solution obtained for problem 1

The cost difference between these solutions and the semi-sharp separation solution is large - the best nonsharp solution almost doubles the cost. Unfortunately, the crude updating of the column costings rules out a reliable comparison between the two on this basis. Satisfaction of the product specifications using nonsharp separations is not as good as that obtained by the semi-sharp separation solution. However, 4 flash vessels with 2 design variables each provides good scope for fine-tuning of solutions both for

better product satisfaction and a reduction in solution costs. To achieve the latter the flashes could be operated at lower pressures, utilising the initial feed pressure to supply the energy for separation rather than the heating utility, thus reducing the heat exchange operating and capital costs. Operation at lower pressure would also reduce the thickness and therefore the capital costs of the flash vessels.

4.3.2 Problem 2

This problem requires the separation of a ternary hydrocarbon feed into two products. The recoveries in the products break the feed component volatility order, i.e. the recovery of the intermediate component in one product is greater than that of the most and least volatile components. Therefore some mixing is required to obtain the products since they cannot correspond to distillation separation outputs.

Feed stream data

Component	flow (kmol/hr)
Propane	100
i-Butane	100
n-Butane	100

Table 4.6: Aggarwal and Floudas Problem 2 - Feed stream component flows

The same assumptions regarding the feed pressure and temperature as for the first problem are made. This results in a value of 298K for the temperature and 13 bar for the pressure.

Product specifications

Again, as with the first literature problem, broad flow bounds are used as the product specifications. Care is taken to ensure that the recovery ordering of the original product set is preserved.

Component	Products (kmol/hr)	
	Product 1	Product 2
Propane	30	70
iso-Butane	50	50
n-Butane	30	70

Table 4.7: Product set data

Specification	Lower bound	Upper bound
Propane flow (kmol/hr)	20.0	40.0
iso-Butane flow (kmol/hr)	40.0	60.0
n-Butane flow (kmol/hr)	20.0	40.0

Table 4.8: Product 1 specifications

Specification	Lower bound	Upper bound
Propane flow (kmol/hr)	60.0	80.0
iso-Butane flow (kmol/hr)	40.0	60.0
n-Butane flow (kmol/hr)	60.0	80.0

Table 4.9: Product 2 specifications

Aggarwal and Floudas’ solution

Aggarwal and Floudas’ best solution is shown in figure 4.4. It consists of two distillation columns performing semi-sharp separations and a bypass of a fraction of the feed stream to the product set.

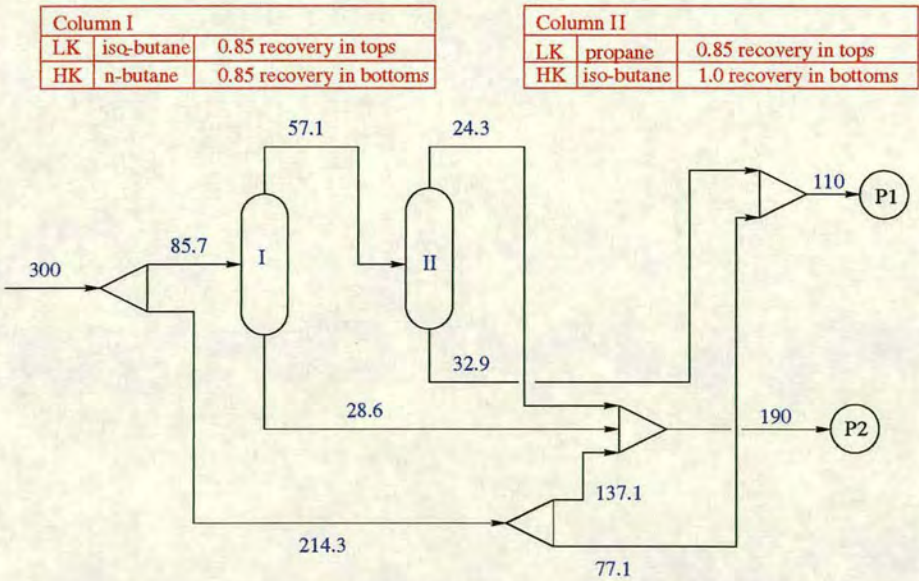


Figure 4.4: Aggarwal and Floudas’ best solution for problem 2

Semi-sharp separation solutions

Given the semi-sharp solution to problem 1 involving a distillation column took over 5 hours, it was decided not to attempt a semi-sharp solution to problem 2 as it needs to consider 2 distillation columns. The MCP synthesis method can generate the solution structure but the greater depth of the search tree combined with the same range of split fraction values will lead to a far greater search time. However, in order to compare the semi-sharp solution with those involving non-sharp separation, the separations in Aggarwal and Floudas' flowsheet were designed and costed on the assumption that they operate at 13 bar. This results in a cost of 335,860 UK pounds for the solution, for a plant life of 1 year. The column designs are shown in table 4.10.

Design variable	Column I	Column II
Light key	iso-butane	propane
Heavy key	n-butane	iso-butane
Light key recovery	0.85	0.85
Heavy key recovery	0.85	1.0
Column pressure (bar)	13 bar	13 bar
Condenser duty (kW)	351	132
Reboiler duty (kW)	188	209
Reflux ratio	1.80	1.48
Number of stages	30	28
Capital cost (UK pounds)	158,540	101,757
Operating cost (UK pounds)	30,938	44,625

Table 4.10: Column designs for Aggarwal and Floudas' solution

Non-sharp separation solutions

The flash model described earlier is also used for this problem. A single temperature level is used and 5 flash vessels are made available. All the splitters are allowed to take on zero and one as the split fraction values. Pruning was activated and the 20 best solutions were requested.

The synthesis program ran for 24.5 minutes and returned 10 solutions containing 4 and 5 flash vessels. Unit design took up 10% of the search time and the search itself the remainder. The solution costs, shown in table 4.11, though more expensive, start

close in cost to the two-column solution. Though each product set has the correct order of recoveries, i.e. recovery of the intermediate component is greater than that of the other 2 components in product 1 and less than them in product 2, the amount of separation between the components in the products is quite small. Given that the flash vessels are restricted to a single design for any feed, there is surely scope for improving both the product satisfaction and the cost of the solutions. These nonsharp solutions deserve further investigation. Figures 4.5 and 4.6 show the best and third best solutions obtained, the flash vessel operating conditions and the component flows in each stream. Solution 2 contains the same flash vessels as solution 1 but adds another flash and performs slightly different blending. The extra flash vessel does not significantly improve separation.

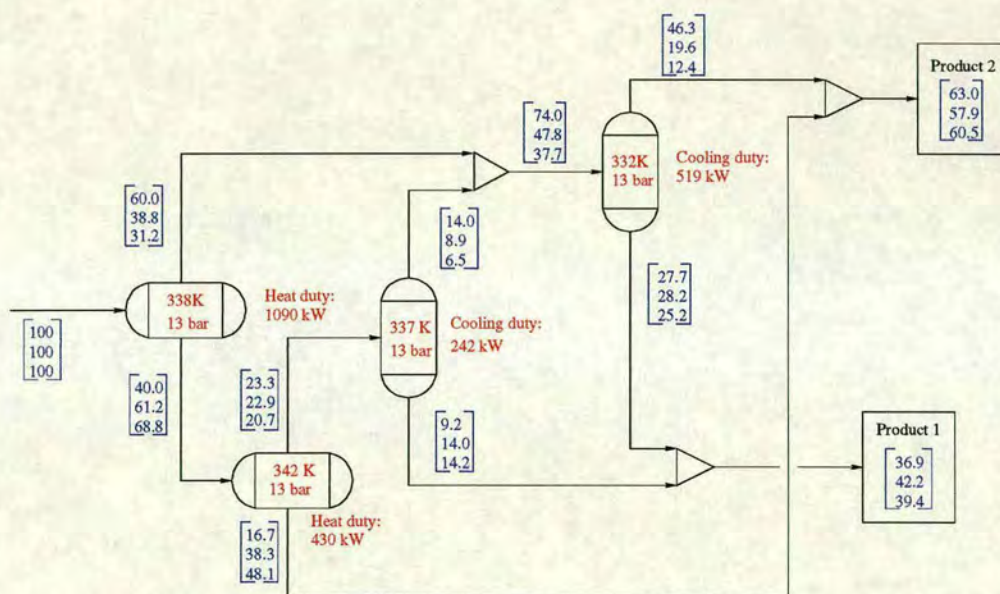


Figure 4.5: Best nonsharp separation solution for problem 2

Sharp and semi-sharp separation solutions

The all-flash flowsheets have poor separation between the components in the final product set because of the difficulty in separating iso-butane and n-butane. There is only an 11K difference between their boiling points at one atmosphere and this is narrowed further for operation at 13 bar. A single semi-sharp distillation column is made available to synthesis to provide more separation between these components. Its

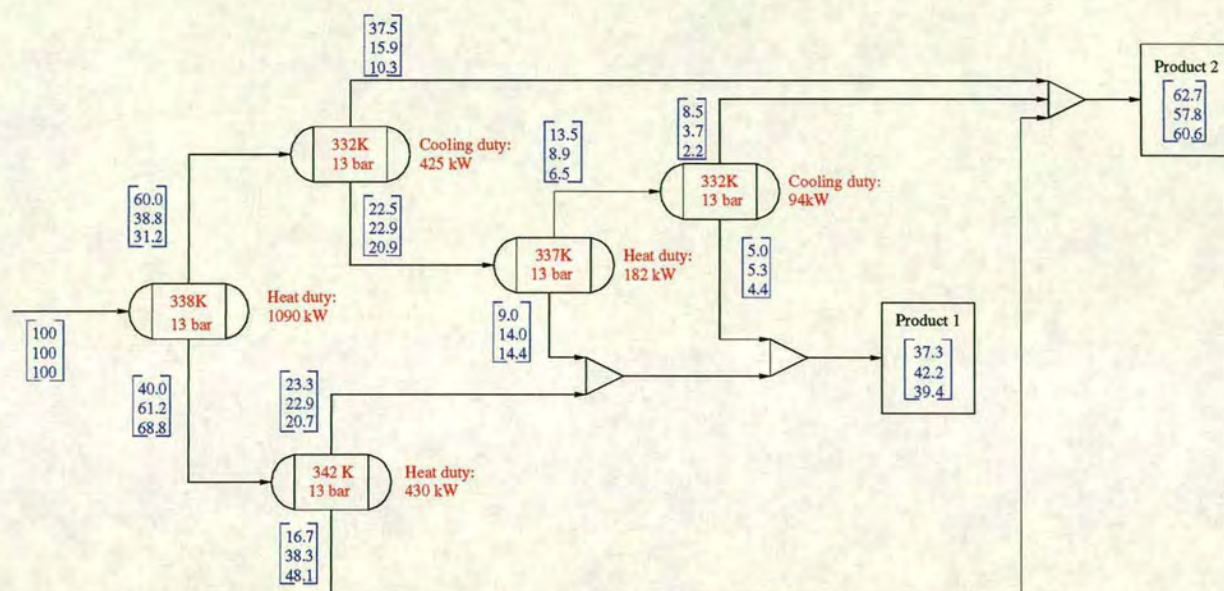


Figure 4.6: Third best nonsharp separation solution for problem 2

Solution	Cost	Relative cost
1	359,111	1.00
2	367,685	1.02
3	381,045	1.06
4	385,795	1.08
5	395,229	1.10
6	408,014	1.13
7	411,002	1.14
8	412,624	1.15
9	414,832	1.16
10	420,058	1.17

Table 4.11: Costs of nonsharp separation solutions to problem 2

light key component is set to be iso-butane and its heavy key to be n-butane and the key recoveries are set to 85% in the top product and bottom products respectively. In order to make use of the same utilities as the flash vessels, the column is operated at 13 bar. 3 flash vessels are made available each operating at 13 bar and with one temperature level.

The MCP method returns 7 solutions, each containing one distillation column and at least 2 flash vessels. The total time taken was 2.63 minutes with the design time taking up 14% of that amount. The solution costs are shown in table 4.12. As expected, the presence of the distillation column produces a greater separation between the feed

components. Though tuning of the flowsheet is still required to improve the satisfaction of the product set specifications, there is now more scope for doing so. The best solution returned (figure 4.7) puts less material through the column separation than Aggarwal and Floudas' best solution and requires only one more heat exchanger. Its structure suggests that a staged pressure drop could be used to exploit the initial high feed pressure such that streams are flashed at their input temperature, thus reducing the heat duties across the sequence of flash vessels. If a significant reduction in nonsharp separation costs can be achieved, this solution will compete well with the 2-column solution.

The second best flowsheet (figure 4.8) uses only 2 flashes but puts more material through the distillation column. Subsequent solutions (table 4.12) also put more material through the column separations either by placing the column earlier in the flowsheet or by blending outputs from flashes to form its feed.

Solution	Cost	Relative cost
1	510,383	1.00
2	516,833	1.01
3	517,298	1.01
4	528,630	1.04
5	550,602	1.08
6	630,141	1.23
7	717,563	1.41

Table 4.12: Costs of semi-sharp, non-sharp solution set to Problem 2

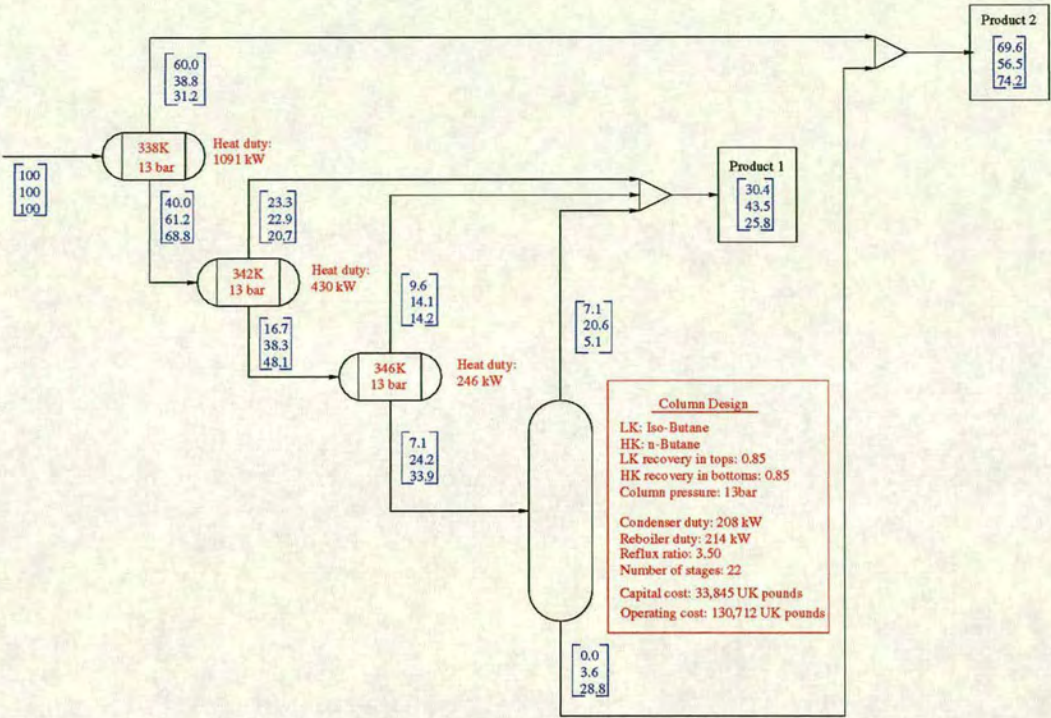


Figure 4.7: Best solution with single distillation column

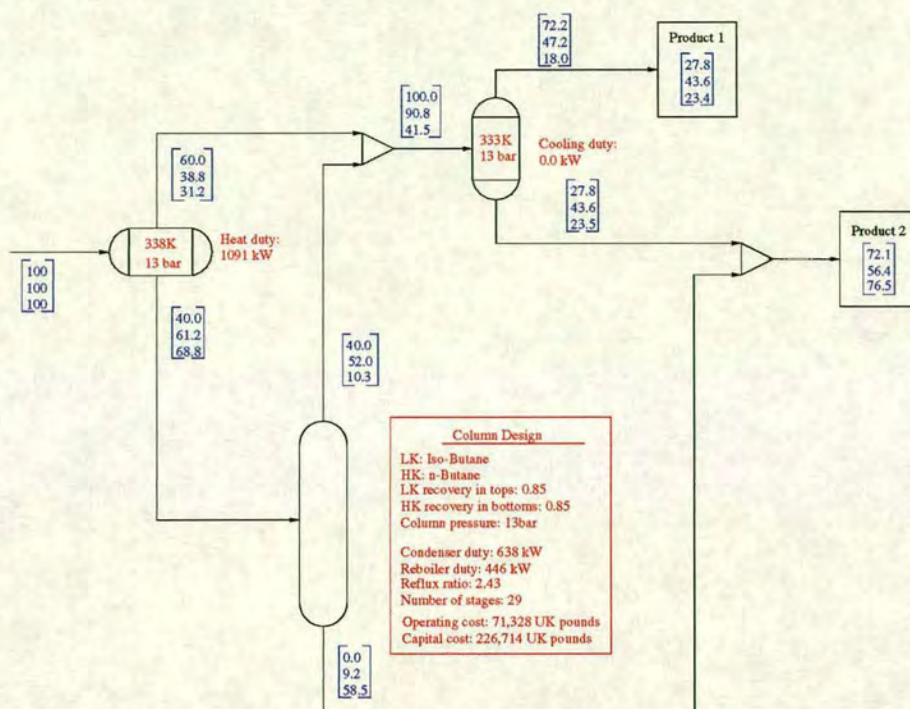


Figure 4.8: Second best solution with distillation column

4.4 Oil stabilisation problem

The aim is to separate a two-phase feed containing twelve hydrocarbon components into specified multicomponent gas and liquid products. A vapour pressure specification is placed on the liquid product and flash vessels are to be employed as separators. This is a multicomponent product problem of industrial scale and interest, utilising an effect specification and nonsharp separators. The existing literature methods, except perhaps for the nonlinear programming approaches, would be unable to attempt this problem.

4.4.1 Problem input

Feed stream data

The feed to the problem is composed of 12 hydrocarbons. Methane and decane are the most volatile and least volatile components respectively and are also the dominant components in the mixture. The other components between these extremes range from approximately 1% to 10% by molar composition.

Flow (kmol/hr)	5200
Temperature (K)	323
Pressure (bar)	29.5
Composition (mole fraction)	
Methane	0.3588
Ethane	0.0773
Propane	0.0980
i-Butane	0.0124
n-Butane	0.0516
i-Pentane	0.0166
n-Pentane	0.0241
n-Hexane	0.0319
n-Heptane	0.0439
n-Octane	0.0468
n-Nonane	0.0378
Decane	0.2008

Table 4.13: Feed stream data

Product specifications

The liquid product specifications (table 4.14) contain an effect product specification in the form of vapour pressure bounds on the liquid product. The gas product (table 4.15) is required to be delivered at 60 bar and the specification is set to accept stream pressures close to that value. For the liquid product the vapour pressure specification ensures that the product is of the desired phase. To achieve this for the gas product an explicit stream phase specification is added. For each product broad flow bounds obtained from a sample solution to this problem are imposed. This prevents streams with low flows, but which satisfy the other specifications, being accepted as product streams. The internal blending operation will gather such streams together.

Specification	Lower bound	Upper bound
Flow (kmol/hr)	1500	5200
Vapour pressure (bar), evaluated at 310.8K	0.0	0.817

Table 4.14: Liquid product specifications

Specification	Lower bound	Upper bound
Pressure (bar)	59	61
Flow (kmol/hr)	2500	5200
Stream phase	gas	gas

Table 4.15: Gas product specifications

Final inputs

The synthesis method was given the feed stream and asked to locate the 20 best solutions which satisfy the product specifications. The flash unit was available (limited to 4 instances) as was the compressor unit (limited to 2 instances). The limits on the unit instances are useful for keeping the search procedure relatively short. The design variable settings for each unit are shown in tables 4.16 and 4.17. Finally, the 20 best solutions were requested, pruning was activated and a plant life of 1 year specified. All splitters in the flowsheet were allowed to take on values of zero or one for the split fraction.

The value chosen for the c_p/c_v ratio is that of methane (1.3) [31] as it is the dominant component in the vapour phase.

Variable	Pressure (bar)	Temperature (K)
Lower bound	1.01325	283
Upper bound	29.5	373
Discrete levels	3	1
Distribution	logarithmic	uniform

Table 4.16: Flash design variable data

Variable	Outlet pressure (bar)
Lower bound	60
Upper bound	60
Discrete levels	1
Distribution	logarithmic

Table 4.17: Compressor design variable data

Results

The 20 best solutions generated by the method include variations on solutions with 2 to 4 flash vessels. All solutions used the two available compressors. The best solutions for each different type of topology are shown in figures 4.9 to 4.13. Associated design information is attached to each unit in the flowsheet. The costs of the 20 best solutions, their costs relative to the best solution and the compressor costs are given in table 4.18.

The time taken to complete the search was just over 3 hours of which the unit design time took up 43.5% of the total time taken and the search time the remainder.

Solution	Cost (UK Pounds per year)	Relative cost	Compressor cost
1	8,276,727	1.00	7,066,131
2	8,304,885	1.00	7,073,234
3	8,334,988	1.01	7,134,360
4	8,337,279	1.01	7,136,877
5	8,337,422	1.01	7,127,217
6	8,337,662	1.01	7,136,490
7	8,338,109	1.01	7,137,648
8	8,343,195	1.01	7,137,906
9	8,343,591	1.01	7,138,302
10	8,355,582	1.01	7,215,589
11	8,455,761	1.02	6,945,724
12	8,474,163	1.02	7,024,778
13	8,482,624	1.02	6,966,924
14	8,798,579	1.06	7,261,378
15	9,047,072	1.09	7,277,189
16	9,052,990	1.09	7,842,394
17	9,085,783	1.09	7,854,132
18	9,215,929	1.11	7,730,227
19	9,220,253	1.11	7,770,868
20	9,254,891	1.12	8,114,898

Table 4.18: Costs of oil stabilisation solutions

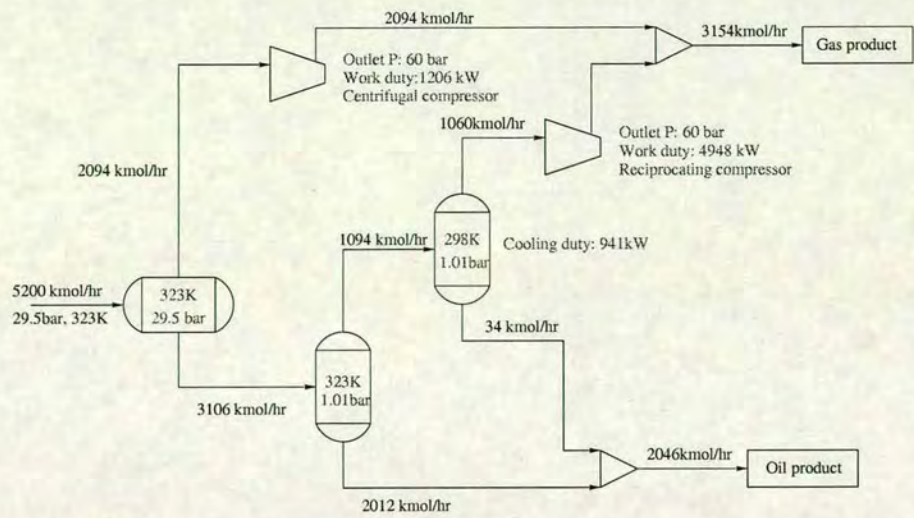


Figure 4.9: Oil stabilisation solution 1

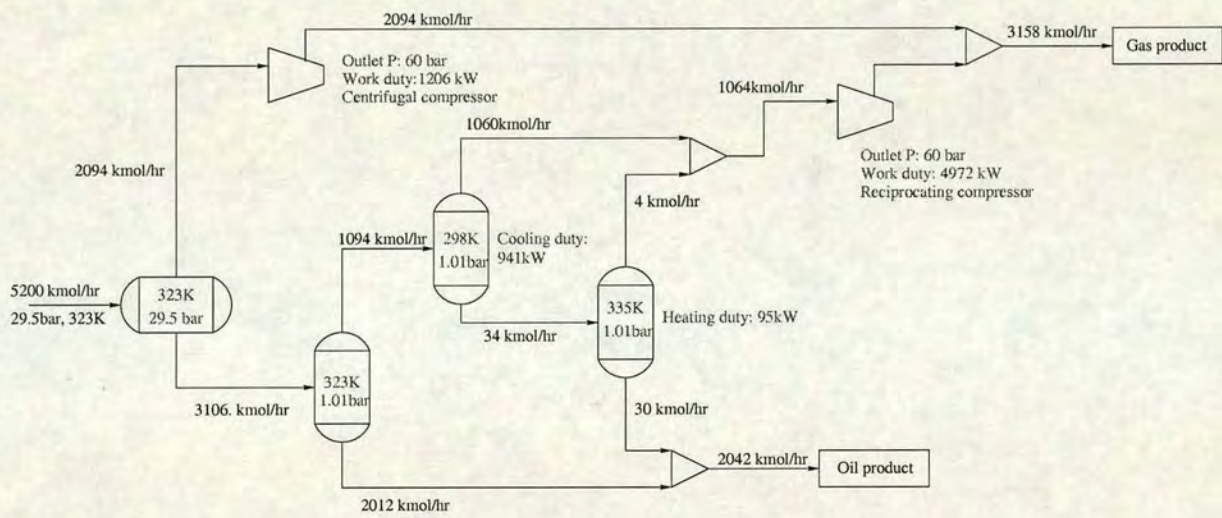


Figure 4.10: Oil stabilisation solution 2

The relative differences in costs between each solution are not large (though the absolute difference the first and last solutions is almost 1 million pounds). This is due to the cost of the second compressor in each flowsheet which has a duty in the region of 5000kW and dominates the solution costs at about 5million pounds for each solution. The operation of this compressor is not realistic - its compression ratio (ratio of outlet pressure to inlet pressure) is always 60, resulting in the need to use a reciprocating compressor instead of the cheaper centrifugal variety. The latter can be used for compression ratios of 5 or under and for work duties at or below 30MW. These conditions can be met if a staged increase in pressure is carried out, i.e. through the use of a chain of centrifugal

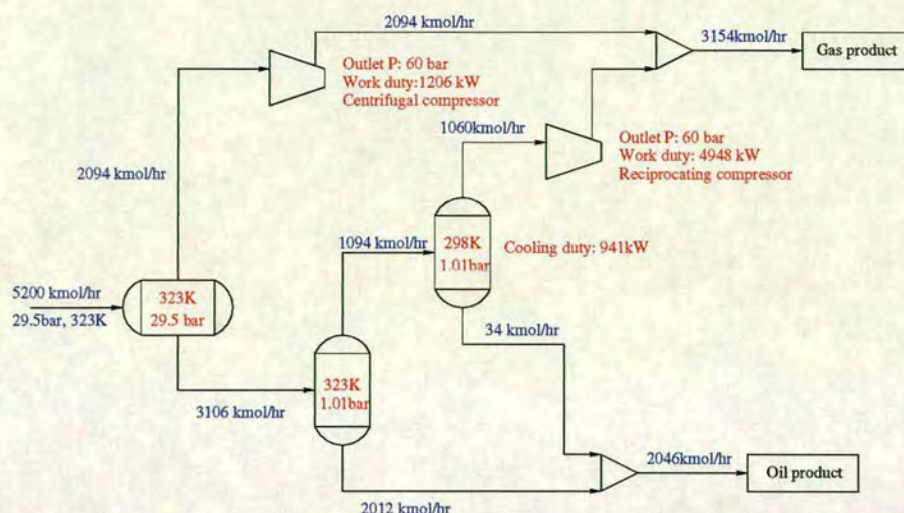


Figure 4.9: Oil stabilisation solution 1

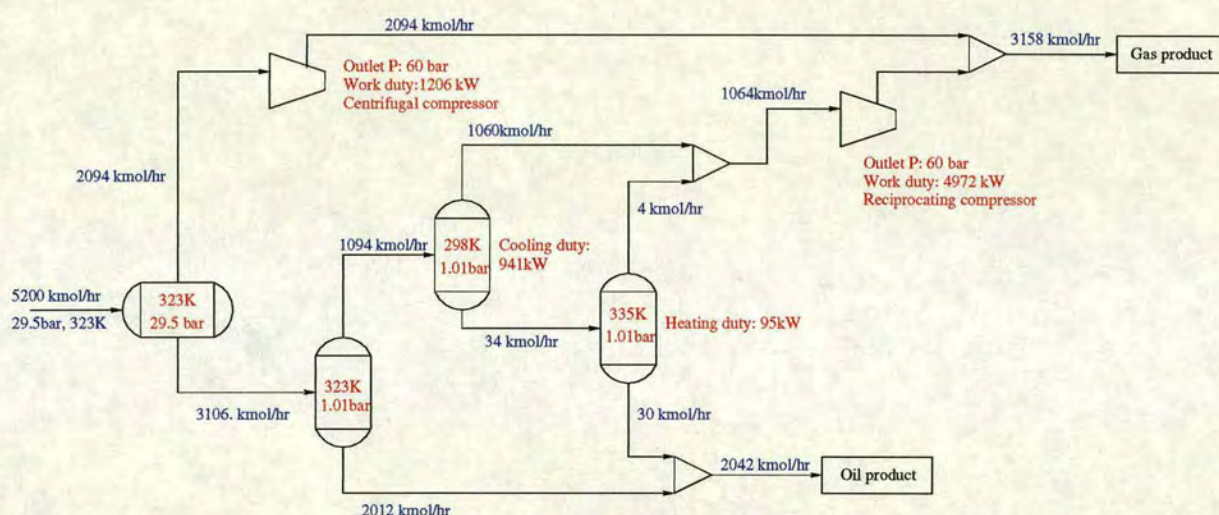


Figure 4.10: Oil stabilisation solution 2

The relative differences in costs between each solution are not large (though the absolute difference the first and last solutions is almost 1 million pounds). This is due to the cost of the second compressor in each flowsheet which has a duty in the region of 5000kW and dominates the solution costs at about 5million pounds for each solution. The operation of this compressor is not realistic - its compression ratio (ratio of outlet pressure to inlet pressure) is always 60, resulting in the need to use a reciprocating compressor instead of the cheaper centrifugal variety. The latter can be used for compression ratios of 5 or under and for work duties at or below 30MW. These conditions can be met if a staged increase in pressure is carried out, i.e. through the use of a chain of centrifugal

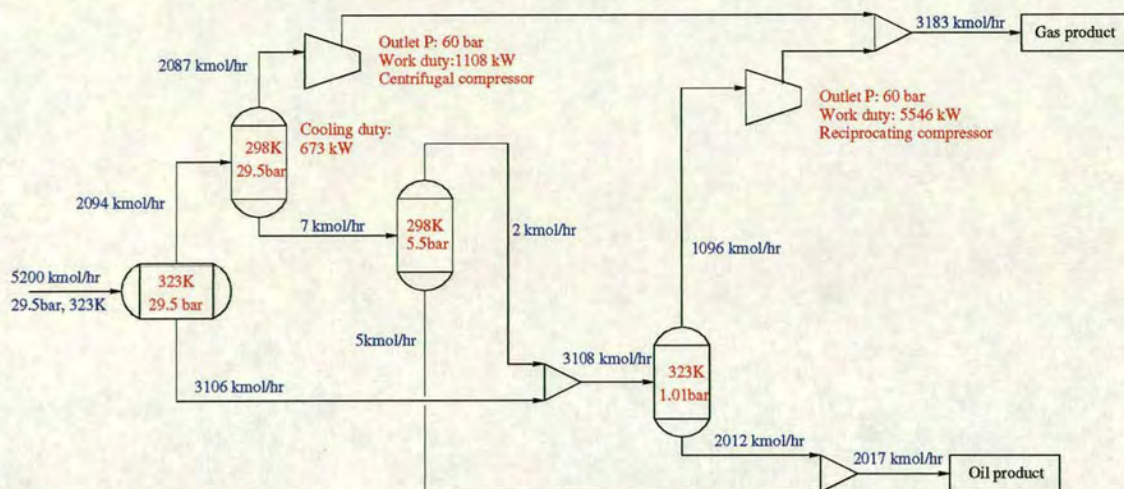


Figure 4.11: Oil stabilisation solution 3

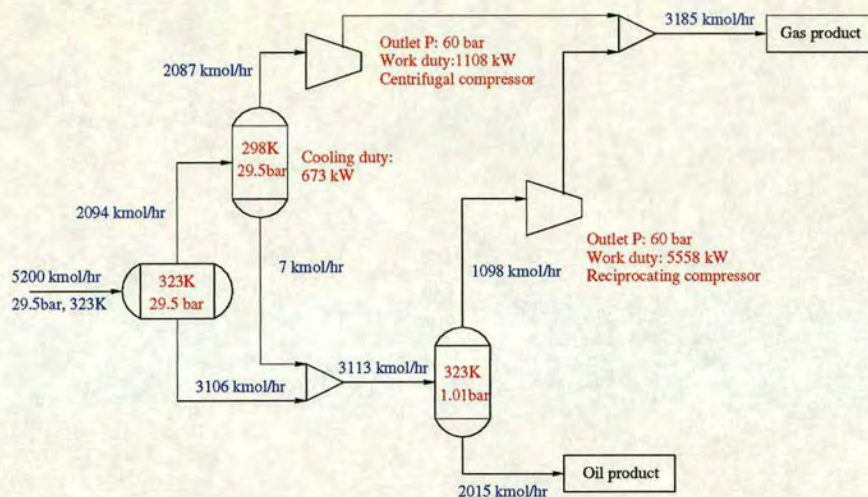


Figure 4.12: Oil stabilisation solution 5

compressors.

More cost-effective compression arrangements can be searched for by increasing both the number of compressors available and the number of outlet pressure levels for each compressor design. However, adding these to the original problem would result in a very large increase in the search space and therefore in the solution time. Instead, compression is treated as a separate problem which takes the compressor inlet streams from the existing solutions as feeds.

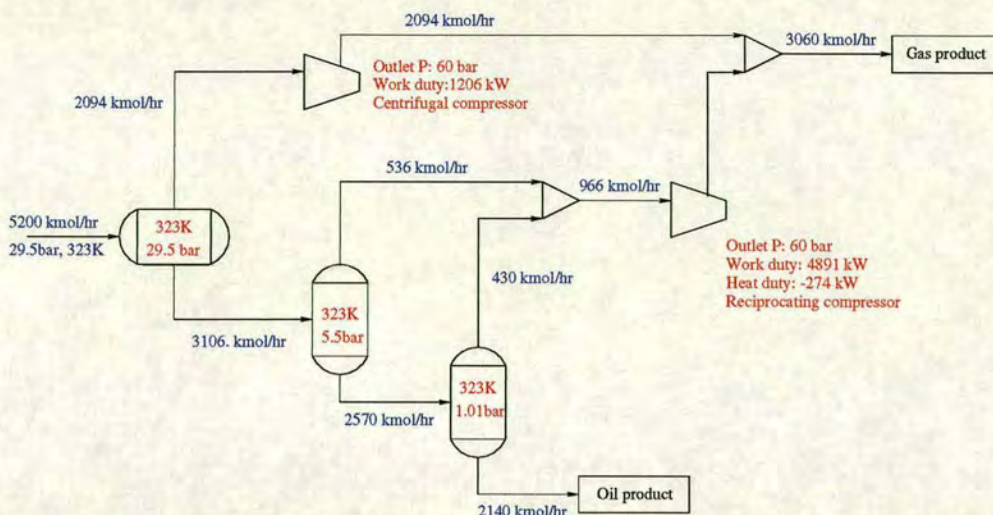


Figure 4.13: Oil stabilisation solution 12

Compression problem from solution 1

This problem has 2 feed streams. 5 compressors are made available, each with 4 discrete pressure levels. Instead of using a discretisation formula to set the pressure values, the SETDVAR (see section A.2.2) routine of the compressor model was altered to provide the following outlet pressure values: 3 bar, 9 bar, 29.5 bar and 60 bar. The aim is to provide the opportunity for the lower pressure feed stream to be compressed to 29.5 bar, then blended with the feed stream of that pressure, followed by compression of the resulting mixture to 60 bar. Going from 1.01325 bar to 29.5 bar using only centrifugal compressors requires at least 3 stages and the compression ratio was kept approximately equal for each stage. The synthesis program was asked to return the 20 best solutions.

The best solution returned is shown in figure 4.14 and is 1.87 million pounds cheaper than the original total compression costs of 7.07 million pounds. Taking this saving into account the total plant cost is now 6.41 million pounds. The second best solution compresses the higher pressure feed to 60 bar, compresses the lower pressure feed stream to the same value over 4 stages and then blends the two, resulting in a marginally higher cost. The costs of the third and following solutions jump due to the use of reciprocating compressors involving compression ratios greater than 5.

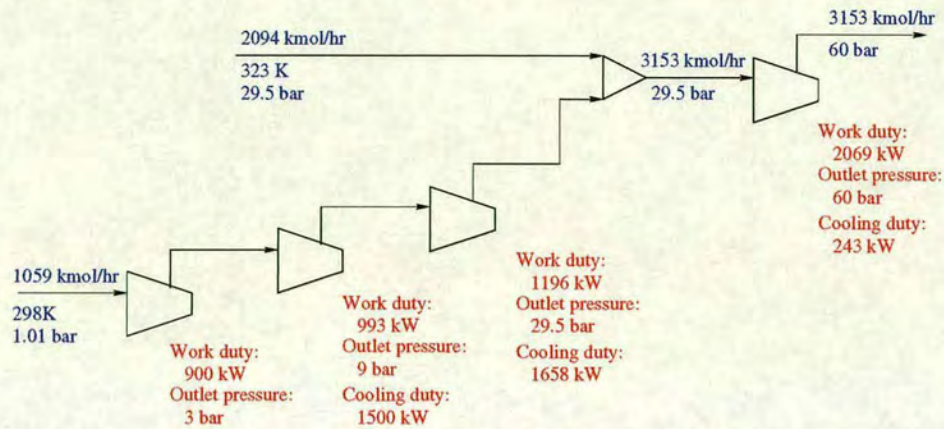


Figure 4.14: Best compressor design for solution 1

Compression problem from solution 12

This solution carries out a staged pressure drop across 3 flash vessels in the flowsheet - from 29.5 bar to 5.5 bar to 1.01325 bar, but the higher pressure in the second vessel is lost during blending with the vapour outlet from the third. A better compression arrangement could take advantage of the higher average flash pressure and result in a more competitive solution. This compression problem has 3 feed streams. 6 compressors are made available (1 more for the extra feed) and use the same number of levels and values as the previous problem. Figure 4.15 shows the best solution to this problem.

Compression costs for solution 12 are now 4.45 million pounds, instead of the original 7.02 million, a saving of 2.57 million pounds which results in a total plant cost of 5.9 million. This beats the adjusted cost of solution 1 by a clear half a million.

Recosting the original solutions

It is clear from the comparison between solution 1 and solution 12 that better compression solutions favour the flowsheets whose flash vessels operate at higher average pressures. Each of the remaining structurally different flowsheets (solutions 2, 3, and 5) provide the inputs for a set of compression problems. Solutions 11 and 13 are also

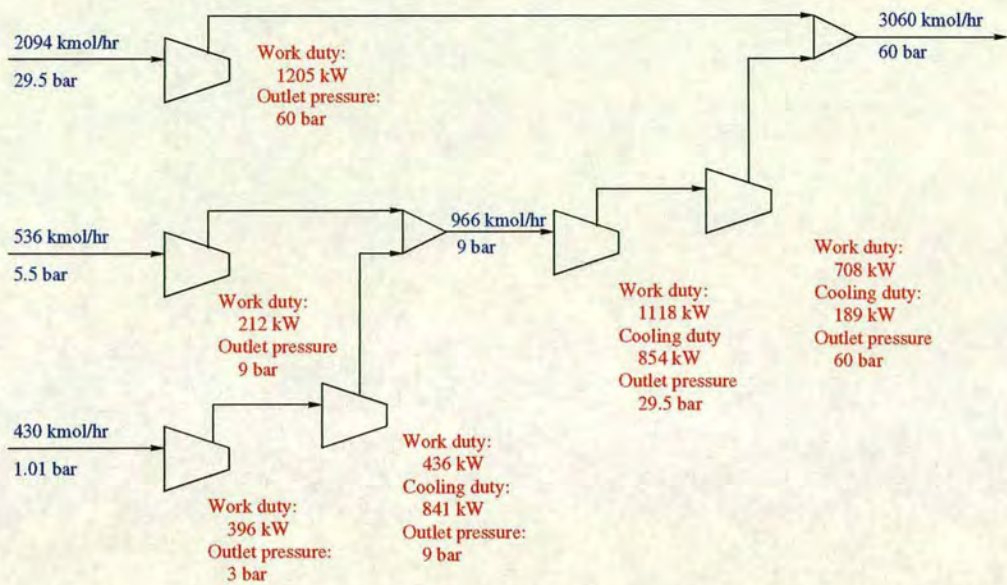


Figure 4.15: Best compressor design for solution 12

useful candidates for a better compression solution since they have lowest vapour feed flows to the compressors of any of the original solutions. Up to 5 instances of compressor units are allowed for each of the problems attempted, the 4 pressure levels for the solution 1 problem used and the 20 best solutions are requested. Table 4.19 shows the full plant solutions using the best returned compressor arrangement, ranked in order in increasing final cost.

Solution	Original compressor costs	Final compressor costs	Final solution cost
12	7,024,778	4,454,166	5,903,550
1	7,066,131	5,197,258	6,407,854
2	7,073,234	5,197,258	6,427,988
13	6,996,924	4,967,378	6,453,079
11	6,945,724	5,147,084	6,657,120
3	7,134,360	5,838,308	7,038,946
5	7,137,217	5,839,170	7,039,374

Table 4.19: Improved compression oil stabilisation solutions

Solution 12, being the only solution with a staged pressure drop across the flash vessels, comfortably beats the other solutions. Solutions 1 and 2 remain competitive with respect to the rest of the solution set but far less so with respect to solution 12. However, optimisation of the flash operating conditions for the other flowsheets could yield the staged pressure drop which benefits solution 12. As can be seen from the table, solutions 13 and 11 do benefit from the improved compressor arrangements in

that they are now ranked ahead of the original solution 3 in terms of cost.

4.4.2 Discussion of results

The results obtained demonstrate that the MCP synthesis method presented is capable of returning a more diverse set of solutions than the literature approaches. It can produce the distillation column and bypassing arrangement as well as solutions containing only nonsharp separators with stream blending and solutions with a mixture of semi-sharp and nonsharp separators. Unfortunately, the mismatched column and flash vessel costs do not permit a reliable comparison between the competing solutions on a cost basis. However, the synthesis method has succeeded in fulfilling one of the requirements for early design, i.e. it returns solutions which can be considered for the next, more rigorous stage of design.

The oil stabilisation problem is the type of problem for which the MCP synthesis method was originally designed, i.e. it contains an effect product specification and requires the use of nonsharp separators. The large number of components has little effect on the working of the method as the generation of the synthesis alternatives does not take this quantity into consideration. The solution set is filled to its specified size of 20 and the 161 solutions were encountered during synthesis even with pruning turned on demonstrating that, for this problem, a large solution space exists for flowsheets involving nonsharp separators and stream blending.

The comparative ease with which problems are specified and modified is a feature of the method and encourages experimentation on the part of the designer. He/she must supply the feed, product specifications and unit models and from this basis the design alternatives are generated. Consideration of the limiting separation between n-butane and iso-butane for the second literature problem resulted in the addition of a distillation column to the set of available separation units. The new unit model and its design variable information must be provided but no more information is required by the synthesis method for the inclusion of the unit into the design alternatives. This flexibility is again demonstrated in the breakdown of the oil stabilisation problem when

searching for better compression arrangements. Using the original solution outputs as inputs to compression problems allows a significantly more compressors to be made available with the result that better solutions are located.

The MCP synthesis method has fulfilled the aims for preliminary design. Its greater exploration of stream space returns a diverse set of solutions and the simultaneous generation and searching of process alternatives combined with the straightforward problem setup reduces designer workload and encourages experimentation. As a result, the designer can bring forward both a set of feasible solutions and a better understanding of the problem to the next, more detailed stage of design.

Chapter 5

Nonideal Separation Synthesis - Background

5.1 Introduction

The aim of this chapter is to provide the background information for nonideal separation process synthesis. This involves a description of nonideality, the tools for analysis of nonideal systems and the manner in which it affects vapour-liquid equilibrium separations. Existing synthesis methods are reviewed and possible contributions to the field are presented.

5.2 Nonideal mixtures and azeotrope formation

Vapour-liquid equilibrium for ideal mixtures is described by Raoult's Law

$$y_i P = x_i P_i^{sat} \quad (5.1)$$

where y_i is the mole fraction in the vapour of component i , P is the total pressure, x_i is the liquid mole fraction of component i and P_i^{sat} is the vapour pressure of component i .

Nonidealities in mixtures occur due to the interaction between the components present. This alters the ideal relationship with the result that mixtures exhibit negative or pos-

itive deviations from Raoult's Law, i.e. for a fixed temperature, the total pressure exerted by the system is either less than or greater than the value given by the above equation. The separability of components (as measured by relative volatilities) is no longer a function of the temperature and the individual components but is also dependent on the given composition and pressure. This behaviour can be adequately described by:

$$y_i P = x_i \gamma_i P_i^{sat} \quad (5.2)$$

where γ_i takes into account the nonidealities in the liquid phase. Values of $\gamma_i < 1$ correspond to negative deviations and values > 1 correspond to positive deviations.

When the deviations become sufficiently large relative to the difference between the pure-species vapour pressures, the $P - x$ surface exhibits a minimum or a maximum. The $P - y$ surface also displays a minimum or a maximum at the same point. A boiling liquid of this composition produces a vapour of the same composition, and the liquid therefore does not change in composition as it evaporates. No separation of such a constant-boiling solution is possible by distillation. The term *azeotrope* is used to describe this state.

5.3 Representation of Nonideal Systems

Nonideality renders equilibrium relationships more complex, resulting in the creation of distillation zones and boundaries. Therefore, it has a strong impact on separation feasibility. To facilitate the design of feasible separations, representations of nonideal systems have been developed.

A *vapour-liquid residue curve* is employed in the analysis of ternary and, to a lesser extent, quaternary systems. It is constructed by tracing the composition of a simple batch distillation in time (Doherty and Perkins [9]). By fitting curves to the composition profile of a continuous column at total reflux, Stichlmair [40] created the concept of a *distillation line*. Residue curve or distillation line maps of a given system can be developed, providing useful information on composition dependence in vapour-liquid

equilibrium. Both representations supply the same qualitative information. Distillation lines are more directly applicable to columns, since they correspond to total reflux operation while residue curves approximate the liquid composition profiles at total reflux. Figure 5.1 is a sketch of the distillation line map for the acetone, toluene, isopropyl-alcohol (IPA) system. The various labelled features of the map are defined below.

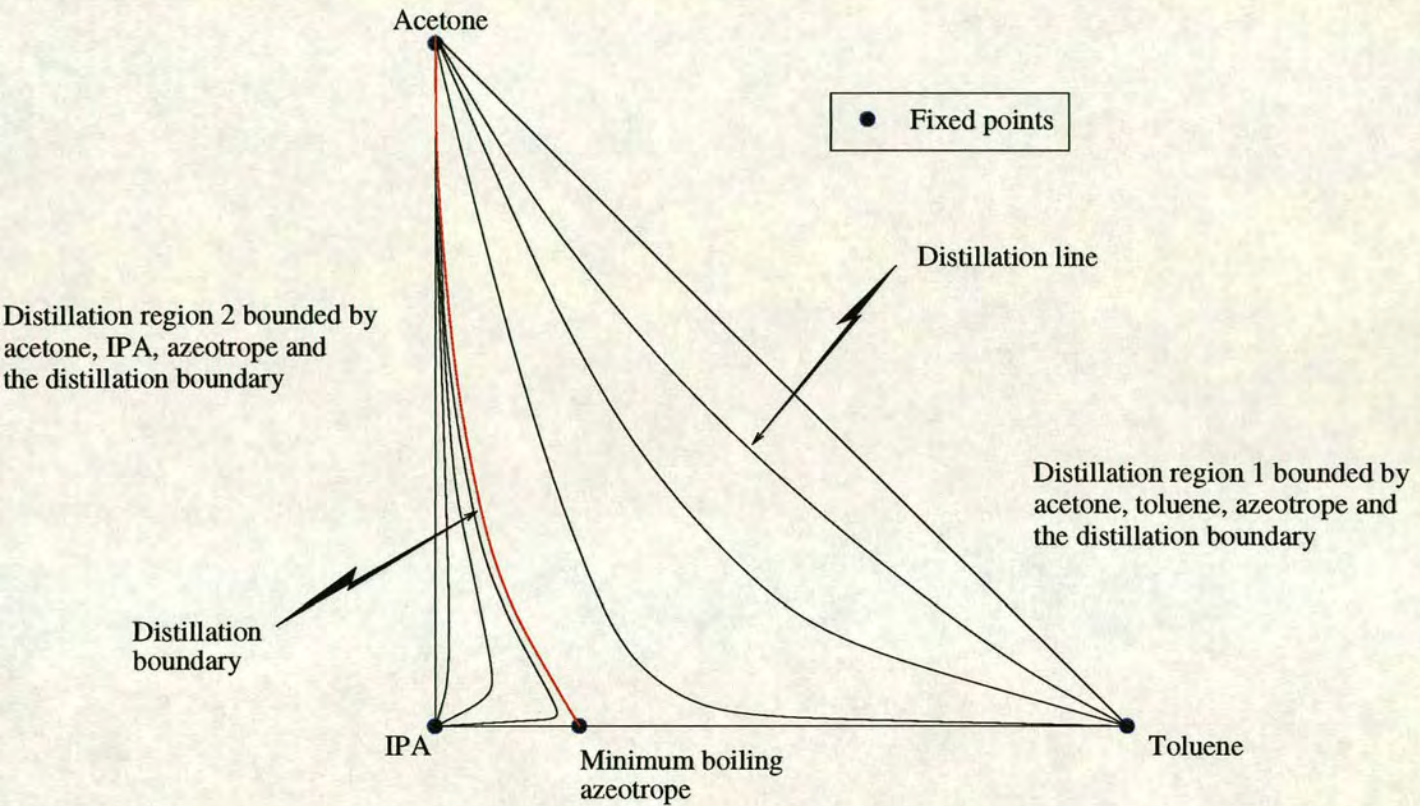


Figure 5.1: Acetone-IPA-Toluene distillation line map

Fixed points

The singular or fixed points of residue curve maps are points in composition space where the driving force for change in the liquid composition is zero. Therefore, fixed points correspond to pure components and azeotropes. There are three classes of fixed point :

- A stable fixed point is one which all local distillation lines converge on. It is the end point the distillation lines. Acetone is the stable fixed point of the example

system.

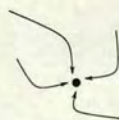


Figure 5.2: Stable fixed point

- When all local distillation lines emanate from a fixed point it is unstable. It is the start point of all local distillation lines. Toluene and isopropyl-alcohol are the unstable fixed points of the example system. Each distillation line has a single unstable fixed point and single stable fixed point, its start and end respectively. These are the *terminal points* of the distillation line.



Figure 5.3: Unstable fixed point

- Saddle fixed points have intermediate boiling points and have lines which converge on them but then continue towards a stable node. The minimum boiling azeotrope formed by isopropyl-alcohol and toluene is the saddle point on the example distillation line map.

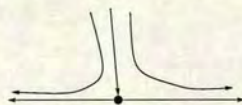


Figure 5.4: Saddle fixed point

Distillation Region

A distillation line map may be divided into different *distillation* regions. In each distillation region, a distillation line map has one unstable fixed point and one stable fixed point, i.e. all lines start and end at those points, making them the most volatile and least volatile species respectively in that region. The terminal points characterise a region.

Distillation Boundary

When a distillation line map possesses two or more different regions, those regions are separated in composition space by distillation boundaries. A distillation boundary is the limiting distillation line between two neighbouring regions. Distillation lines in either region approach the boundary closely but do not cross it. Curvature in the boundary is common and reflects the selectivity with which components modify each others volatilities. A straight line boundary implies no such selectivity and is rare in the case of nonideal mixtures.

5.4 Feasible separations in nonideal ternary systems

The basic difference between the design of separation sequences for ideal and nonideal mixtures is that the product distribution for the latter depends on the distillation region in which the feed lies. Hoffmann [17] lays out two conditions for feasible separations. Firstly, the overall mass balance must be satisfied, easily checked knowing the feed and product flows and compositions. Secondly, there must be at least one path of energy balances and equilibrium relationships describing each stage of the distillation column. For a feasible separation at total or infinite reflux this condition is described by a distillation line connecting the top and bottom product compositions.

In ideal systems, the region of feasible separations is bounded by the distillation line through the feed F and the straight lines through F and the terminal points of the distillation lines (figure 5.5). The distillation line through the feed limits separation since, to arrive at products outside it would result in the intersection of distillation lines at points other than the terminal nodes of the feed region. For nonideal systems, the same principle applies for each distillation region. Unlike ideal systems, one or both of the terminal points may be an azeotrope (figure 5.6). Also, the separation may be further bounded by the presence of a distillation boundary (figure 5.7). It is also possible for a separation to cross a distillation boundary. Feasibility requires that only the products are connected by a distillation line. Therefore, the products may lie in a

different region to the feed. When the feed lies in the composition envelope bounded by the distillation boundary and the straight line approximation of that boundary, the products can lie on the opposite side of boundary to the feed. The product regions are bounded by the distillation boundary and by the straight lines through the feed and boundary fixed points as figure 5.8 illustrates.

Laroche et al. [20] have demonstrated that there exists a maximum reflux ratio for nonideal separations, beyond which separation starts decreasing. In some cases, at the limiting condition of operation at infinite reflux no separation occurs. Therefore, distillation line and residue curve maps will overlook some feasible separations. However, their simplicity and speed of calculation makes them a useful tool for determining the feasibility of separation in the first stage of synthesis.

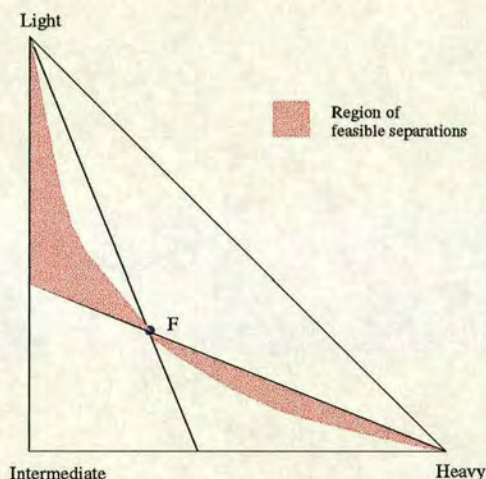


Figure 5.5: Separation region of a ternary ideal mixture

5.5 Entrainers

An entrainer is a substance added to a mixture, whose purpose is to facilitate, or aid the separation of the mixture constituents. After having determined if azeotropes are present in a system, the next step is usually to find entrainers best suited to it. This present work is concerned with the synthesis of a flowsheet for a previously chosen entrainer or set of entrainers. Nevertheless, entrainer selection is an important issue to consider and a brief overview of this area will be provided.

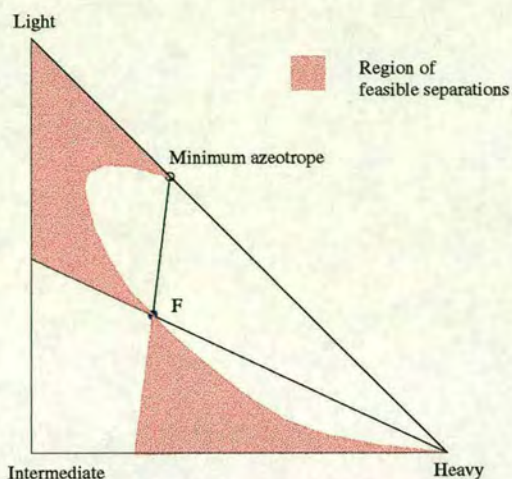


Figure 5.6: Separation region of a ternary nonideal mixture - no distillation boundary

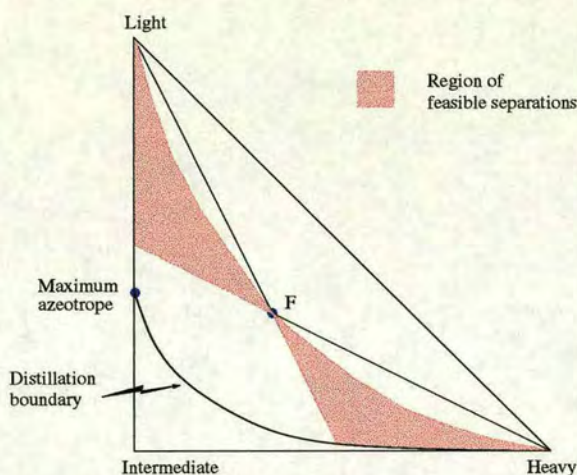


Figure 5.7: Separation region of a ternary nonideal mixture - with distillation boundary

The choice of entrainer has a large impact on the economics of the separation system. Its addition increases the amount of material which the system must handle, increasing capital and operating costs. As a result, the entrainer performance is crucial, i.e. it must selectively modify the volatilities of the system components so as to permit or enhance their separation.

Much effort has gone into methods for the selection of the best entrainers for a process. So far work has concentrated on ternary systems where the knowledge gained has been largely deduced from residue curve and distillation line maps and these have proved an effective tool for entrainer selection. It should be noted that the entrainer is permitted to form azeotropes with the feed components. In that case those azeotropes are the

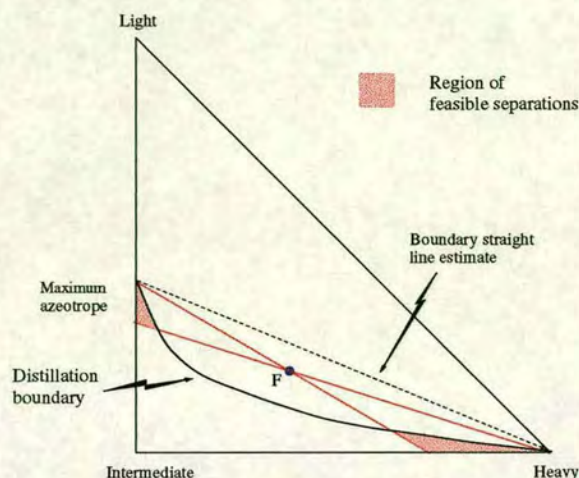


Figure 5.8: Boundary-crossing separation regions

species which facilitate separation, i.e. they fulfill the entrainer function.

Entrainers can be classified according to whether or not they induce boundary-crossing [1].

- Potential entrainers for non-boundary-crossing flowsheets must keep the feed azeotrope components in the same distillation region. As a result, the feed azeotrope is *never* a saddle node [12, 39].
- Potential entrainers for boundary-crossing flowsheets introduce a curved distillation boundary. Feed azeotrope components are in different regions and are stable nodes [41].

These are the basic feasibility requirements and can be tested by constructing the residue curve or distillation line maps for the feed components and candidate entrainers.

5.6 Nonideal Separation Synthesis Methods

Much work has been done on the separate parts of the nonideal separation problem. However, until recently, not many synthesis methods existed which brought all aspects of the problem together. Three such methods are now reviewed.

Wahnschafft et al. [42, 44] present a tool named *SPLIT* for the generation of feasible separation processes. *SPLIT* addresses problems where the major difficulty may be the generation of feasible solutions rather than the pruning of weak alternatives. This makes it suitable for azeotropic synthesis since the limitations on separation imposed by the presence of azeotropes and distillation boundaries leads to difficulty in locating feasible separations. It makes use of a blackboard architecture which brings together different knowledge sources used to derive and rate solutions. These knowledge sources share the common memory “blackboard” which is monitored for patterns indicating the ability for various separation tasks to be executed.

The separation problem is specified by descriptions of the feed and desired products and the system reformulates the problem as a set of binary separation tasks. The separation step is chosen by analysing to see which tasks have yet to be carried out. The separation tasks are posted and the knowledge sources propose the appropriate separation technologies. Two options for the feasible split and the corresponding operating conditions are considered:

- Maximum separation between components with different final destinations.
- Minimum separation of certain streams so that output streams can later be sharply separated into products.

Simulation of the chosen separation technology across its operating range is used to determine the separations available. While this is time-consuming it does not require preliminary knowledge of the distillation boundaries in azeotropic problems. If sharp splits are feasible only the first option is considered. The choice of recycles is also automated for the following cases:

- To facilitate separation, eg. entrainers. An entrainer is added to create a feasible separation and this sets up a mixing goal which the recovered entrainer must satisfy.
- The flowsheet already contains separation steps which lead to the desired separ-

ation. The stream requiring separation is then recycled to the point where the desired separation occurs.

Alternative solutions are compared by combining weighted judgements from each available knowledge source. In essence this is the application of heuristics with the ability to recognise conflicts and trade-offs. Alternative solution paths can be explored in parallel. However, knowledge or the user can focus the search and restrict alternatives.

Malone and Doherty [24] have developed a set of computer tools called MAYFLOWER to aid the synthesis of nonideal separation processes. It provides the user with the capability to check the feasibility of and design individual units for complex mixtures. The procedure requires the specification of the feed composition, product purity and the operating pressure. The feasibility check involves the construction of the residue curve map for the mixture from an appropriate VLE model. If such a model is not available or is incomplete the map can be sketched using the method of Foucher et al. [12]. If the sketch is not enough then the data can be obtained by experiment. Sets of heuristics for the sequencing of units and unit design are presented and cost correlations used to evaluate the design alternatives. The procedure results in a set of comparable least expensive alternatives which are intended for further optimisation.

Bauer and Stichlmair [5] create a superstructure containing all separation alternatives which is put in the form of a mixed-integer nonlinear programming problem and solved accordingly. The superstructure is composed as a sequence of *preferred* separations, such that each column performs a nonsharp split at minimum reflux. This approach allows an upper bound on the number of columns to be determined and results in the inclusion of complex column configurations in the superstructure, eg. prefractionators, coupled columns, side-strippers. The approach is feasible for any number of components. However, the determination of the preferred separation for systems with distillation boundaries requires foreknowledge of the boundaries or some means of detecting them. Without this information the superstructure cannot be constructed. The modelling of separations is rigorous and the objective function for optimisation is the minimum heat demand for a fixed number of stages for each column in the sequence.

The authors intend to develop an objective function using the total annualised cost of flowsheets.

5.7 Contributions to be made

The work done on nonideal systems has led to a solid understanding of how to construct feasible sequences of separators. Distillation line maps and residue curve maps and the information they contain have played a large part in this. However, automated synthesis methods which make use of this information do not exist. MAYFLOWER uses them but it is intended as an aid to synthesis rather than as a synthesis method itself. The superstructure approach of Bauer and Stichlmair requires distillation line map information but does not incorporate it. Wahnschafft's method is very powerful but it is desirable to remove the necessity for simulation as a means to locate feasible separations.

The presence of azeotropes and distillation boundaries imposes limits on separations in nonideal systems. As a result a greater exploration of stream space is required than for pure component product problems in ideal systems. Also, the recycles encountered need not be pure components or azeotropes as there exists a trade-off between recycle purity and the recycle flowrate. Wahnschafft defines a region of composition for recycles based on the preservation of separation tasks within a flowsheet leading to an *effect* product specification. These are features in common with the MCP product problem for which a synthesis method has been developed. The intention is to apply the MCP synthesis method to nonideal separation problems by equipping it with distillation line map information and the ability to handle recycles.

Chapter 6

Nonideal Separation Synthesis

6.1 Introduction

This chapter describes the work undertaken to equip the MCP synthesis method with the means to solve nonideal separation problems. There are three issues to be addressed in order to achieve this. Firstly, unlike ideal/near-ideal separation problems, the identification of the feasible separations required to construct flowsheets is a non-trivial problem. This is overcome through the use of nonideal system information in the form of distillation line maps for the purpose of targetting feasible separations. Second, nonideal separations must be designed and a shortcut model used for this purpose is presented. Finally, solutions to nonideal separation problems require recycles to maximise profits. The approach to the inclusion of recycles, the alternatives available and their integration into the MCP synthesis method is detailed.

6.2 Targetting nonideal separations

The description of the synthesis problem in this work consists of the process stream front and the set of targets which solution flowsheets must satisfy (Chapter 3). This is a very general synthesis problem description and it has facilitated the inclusion of many features which existing MCP synthesis methods do not possess such as splitting and blending, nonsharp separations and effect product specifications. However, it contains

no information on how to solve a given problem, i.e. problem-specific knowledge which can be used to both reduce the search space and focus on certain alternatives. For ideal/near-ideal MCP problems, the lack of such information can result in a large search space. However, the generation of feasible separations required to build up flowsheets is not a difficulty. This is not the case for nonideal problems where the limitations on separations imposed by the presence of azeotropes and distillation boundaries lead to difficulty in locating feasible separations. This problem is addressed through the use of distillation line maps for the analysis of nonideal systems and the location of feasible separations during synthesis. The manner in which this is achieved is described in this section: the construction of the distillation line map for each synthesis problem is outlined, followed by the separations which are targetted. Finally, the algorithm which performs the targetting function during synthesis is presented.

6.2.1 Modelling distillation line maps

Distillation line maps are chosen over residue curve maps because they are more suited to this work. Their determination is straightforward and, since the design model used for nonideal separations (section 6.3) contains a total reflux calculation, the separation limits derived from the distillation line map correspond to those given by the design model. For this work, the emphasis is on the use of distillation line maps rather than on their construction from scratch. Building maps from components and their properties requires the location of azeotropes, distillation boundaries, the determination of the individual distillation regions and the fixed points which comprise them. Work is being done by other authors in these areas [8], [10]. For now it is assumed that the user has the distillation line map information or the means to determine it. He/she supplies the azeotrope data and distillation boundary and region information. This is then structured and made available to the synthesis method. The construct which achieves this is shown in figure 6.1. The description of its implementation and the elements which comprise it are given in Appendix B.

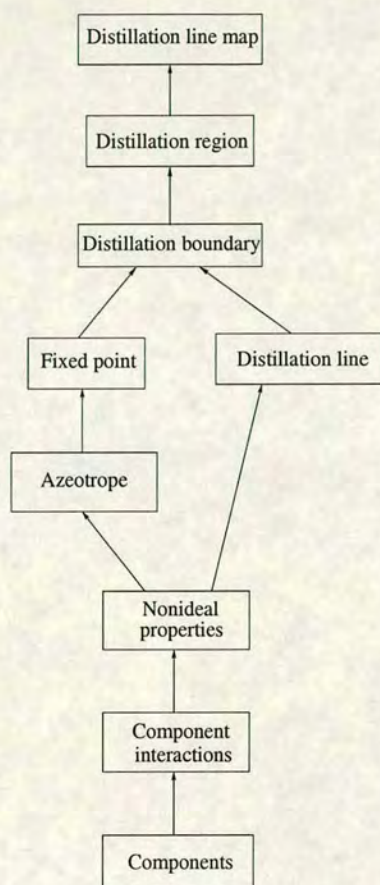


Figure 6.1: Nonideal system information hierarchy

6.2.2 The desired separation alternatives

The existing synthesis methods for nonideal separations [5, 24, 43] have demonstrated that a small set of alternative separations is sufficient to provide good solutions to these problems. Therefore, the separations proposed by those authors are targetted.

Wahnschafft et al. [44] perform an analysis of the feasible separations on each process stream encountered, using simulation to search for maxima in the separations between components and cases where the minimum separation requirements are met. Their analysis, which is based on the representation of separation problems as a set of binary separation tasks, is not used for this work. Instead, the results of that analysis are exploited. The maximum separations correspond to those where one of the outputs is a product stream, i.e. a pure component. The minimum separations produce outputs which can later be separated into the desired products. The equilibrium "conode" sep-

aration used by Bauer and Stichlmair [5] closely approximates the minimum separation of Wahnschafft et al. but is much more straightforward to determine and is therefore used in its place. The resulting separation alternatives made available to different feed streams are now discussed.

For a ternary separation problem four separations are made available. Working in the feed distillation region, the first separation alternative produces the most volatile species as a product, the second the least volatile species. The third alternative is the equilibrium conode separation. In this case the orientation of the separation mass balance line is given by the vector from the feed composition to its bubblepoint equilibrium composition. This results in a separation whose products will lie between those of the first two alternatives. The final alternative is a boundary-crossing separation which accesses a neighbouring region and produces the boundary pure component fixed point as an output. These are the available separations and they are summarised in figure 6.2. Not all these separations are necessary or feasible for the given feed. All the alternatives overlap for a binary feed and as a result only one is declared feasible. The boundary-crossing separation alternative is only feasible if the feed composition lies in the region bounded by the distillation boundary and its straight line approximation (see figure 5.8).

6.2.3 Setting the design specifications for the targetted separations

The distillation line map information assembled for a nonideal separation problem provides the means to identify the region of feasible separations available to a given feed (section 5.4). This information is then used to target the set of desired separations and takes over from design variable discretisation (section 3.4) as the means of generating the separation design alternatives.

The separation design variables are split into two groups: the first is comprised of the separation pressure. This is still set by discretisation as the distillation maps and therefore the region of feasible separations are influenced by its value. The second group consists of the variables used to characterise the separation, i.e. they specify the

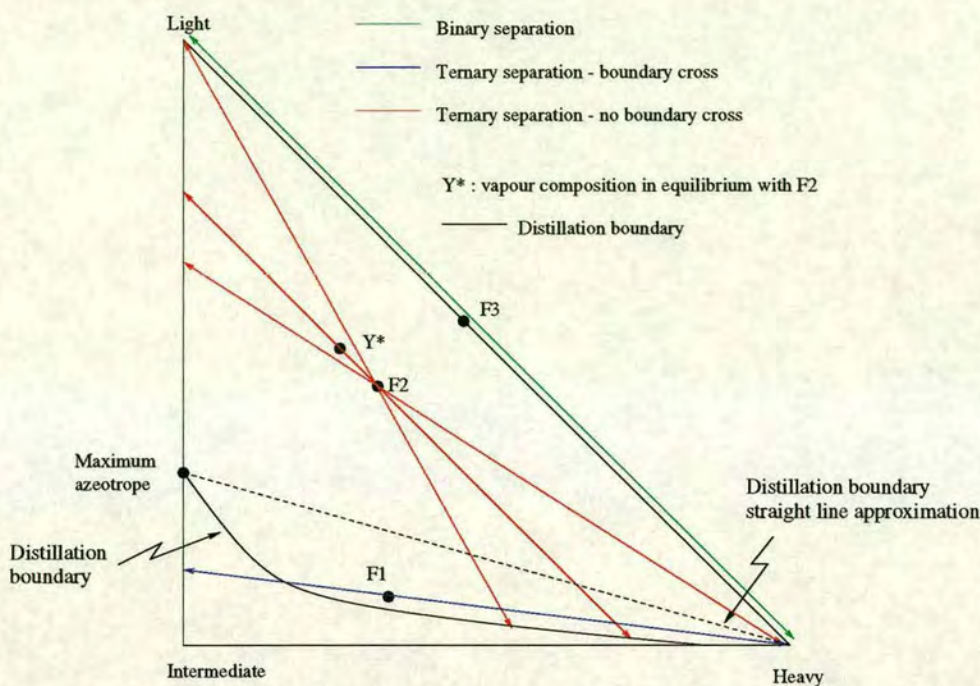


Figure 6.2: Nonideal separation alternatives

component distribution in the outlet streams. Their values are set using a targetting procedure. More detail on the organisation of design variable data for targetting and the integration of targetting into the synthesis method design alternatives can be found in Appendix B.

The algorithm for setting the value of the separation design variables by targetting takes the feed stream, the separation pressure and the desired separation alternative as inputs. First, the feed region is determined. Next, if the boundary-crossing separation is the current alternative its feasibility is tested. The product region is then determined - this is the same as the feed region if a non-boundary-crossing separation is under consideration. If the boundary-crossing separation is desired and if it is feasible, the product region is that which the boundary cross accesses. Finally, the full extent of the desired separation is determined and the separation specifications are set. Each of these steps is now described in more detail.

Inputs to the targetting algorithm

The algorithm requires the feed stream composition, the operating pressure of the distillation column and the current separation alternative. The latter two inputs comprise the design variable set for the nonideal separation. The operating pressure is a discretised variable while the remaining variables make up the targetted group whose values are dictated by the current separation alternative.

Determination of the feed region

The feed region must be known in order to target the separation products. The method for determining this is straightforward:

- Obtain the distillation line through the feed composition.
- Step through the distillation regions in the distillation line map and compare the endpoints of the path with the terminal fixed points of the current distillation region.
- If the path endpoints correspond to the terminal fixed points of the current region then this is the feed region.

Feasibility of a boundary-crossing separation

When the boundary-crossing separation is the current alternative, its feasibility must be tested before the separation can be specified for design. The procedure for testing feasibility is as follows:

- Step through the distillation boundaries of the feed region.
- For the current boundary, check if its path has been determined. If not then do so.

- Check if the boundary is concave with respect to the given region. This test is carried out by obtaining the composition corresponding to the midpoint of the boundary path straight line approximation. The region containing this composition is determined in the same way as for the feed stream composition. If it is the same as the feed region then the boundary is concave. If not, then a boundary-crossing separation is not possible for the given feed composition. Figure 6.3 illustrates.
- Work out if the feed composition is inside the boundary-crossing region. This is the space enclosed by the boundary path and its straight line approximation. A line is constructed from a boundary path node (other than the fixed points of the boundary) to the feed composition. The parametric form of a line is used with the parametric variable taking on values from zero to one. If the intersection of this line with the boundary straight line approximation is outside the bounds of the parametric variable then the feed is inside the boundary-crossing region as shown by figure 6.4.

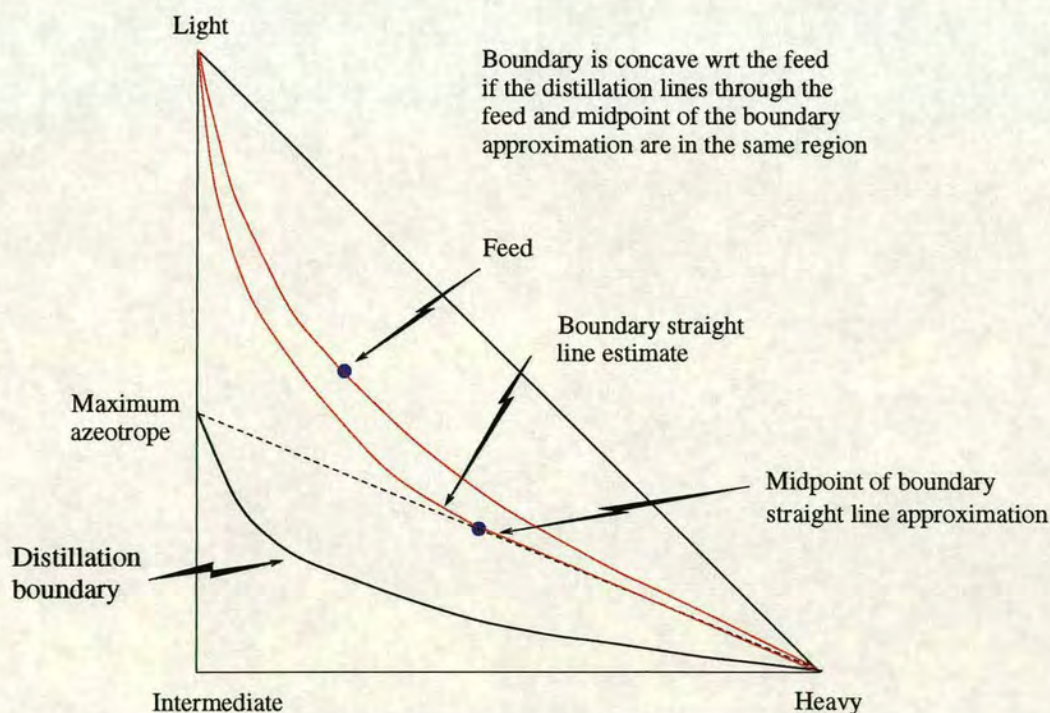


Figure 6.3: Checking if the boundary is concave wrt the feed

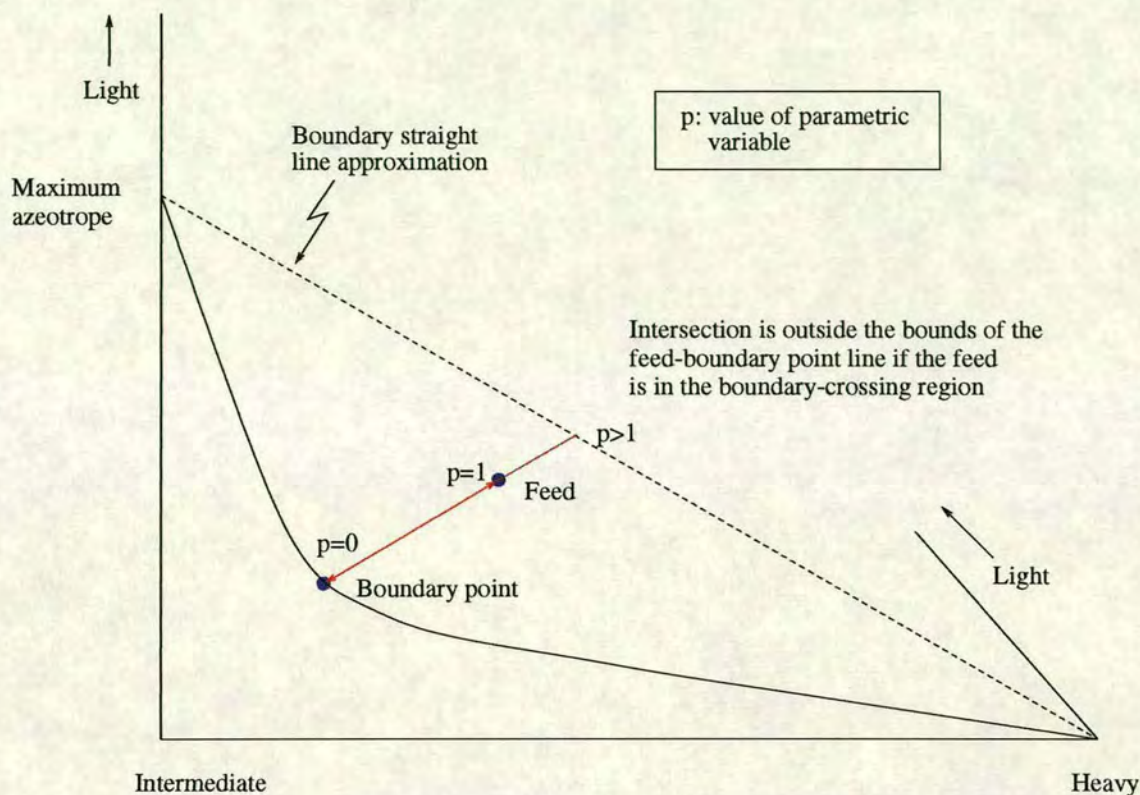


Figure 6.4: Testing if the separation feed is inside the boundary-crossing region

Product distillation region and partial mass balance line

The current design alternative is part of the input to this algorithm. Depending on whether the feed is binary or ternary or whether or not a boundary cross is possible, the appropriate alternative is extracted as a composition. This composition will correspond to the most volatile species, least volatile species, the equilibrium conode or the boundary terminal fixed point. It is used to form a partial mass balance line with the feed stream composition. The distillation region of the separation products is also determined. For boundary-crossing separations this is the region which shares the boundary with the feed region. When a boundary cross is not feasible the product region and feed region are the same.

Determination of maximum extent mass balance line

The separation alternatives considered are pushed to their maximum extent, i.e. until no more separation along the mass balance line is possible. The last step established the orientation of the mass balance line. The task now is to determine the maximum amount of separation along this line. This is carried out as follows:

- Check if the feed mixture is binary or ternary.
- If binary then:
 - Test if there is an azeotrope formed by the two components. This involves searching the user-provided azeotrope data and comparing the constituent components of each binary azeotrope with the given components.
 - Work out which of the two components can be obtained as a pure component by distillation. This is one end of the mass balance line, the azeotrope is the other. Formulate the mass balance line equation.
- If the feed mixture is ternary then the following steps are performed. The equilibrium conode separation shown in figure 6.5 is used as an example.
 - Obtain the intersection of the partial mass balance line, $x_f - y^*$, with the binary edges of the distillation line map. This leads to the separation $T - x_f - B'$.
 - Step through the distillation boundaries of the *product* region and determine the intersection of the mass balance line with them. For the example provided this leads to the separation $T - x_f - B$. If a separation is crossing a boundary, the intersections with that boundary are not determined.

Setting the column model specifications

The separation is now fully defined. At this point the feed stream, operating pressure and the mass balance line are known. The product compositions are extracted from

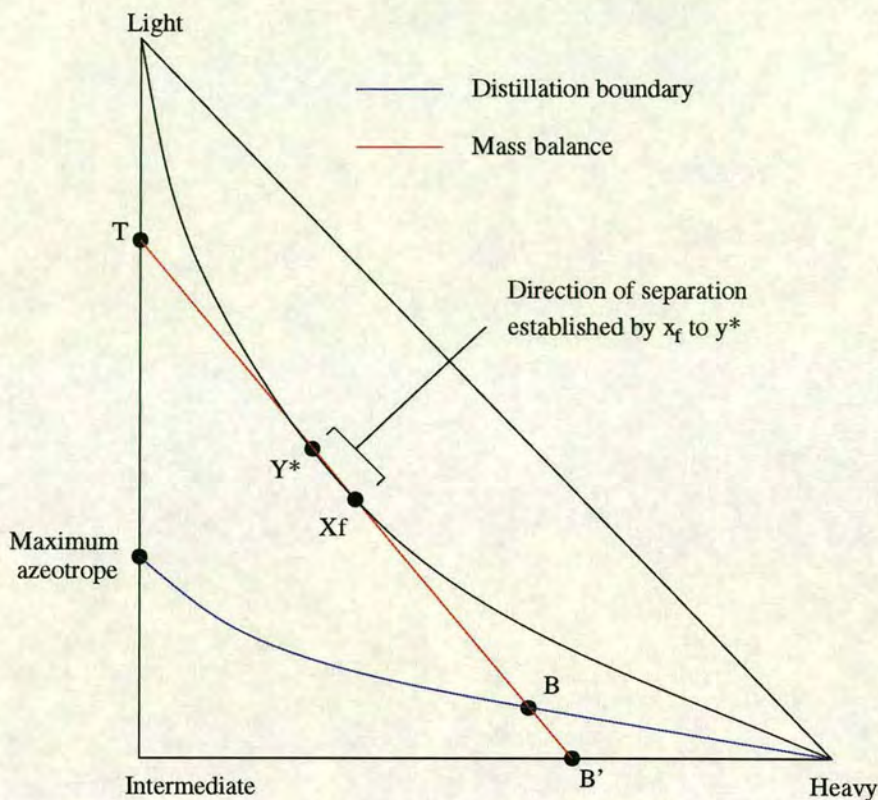


Figure 6.5: Determining the maximum extent of an equilibrium conode separation

the mass balance line equations and combined with the feed stream data to solve for the product component flows. This provides the input for the procedure which sets the column model specifications described in the next section.

6.3 Modelling Azeotropic Separations

The focus of this work is on early design. Therefore, the use of a fully rigorous column model for nonideal separations is not consistent with this aim. Instead a shortcut model originally developed by Senos Matias [34] and improved by Herron [16] was employed.

The shortcut nonideal column model mimics the Fenske-Underwood-Gilliland (FUG) design method for ideal separations but employs rigorous activity-based thermodynamics to model the vapour-liquid equilibrium. This does not affect the solution of the Underwood method for the minimum reflux (provided the pinch is at the feed which is

therefore less robust. A means of determining the anchor product for such separations would be useful.

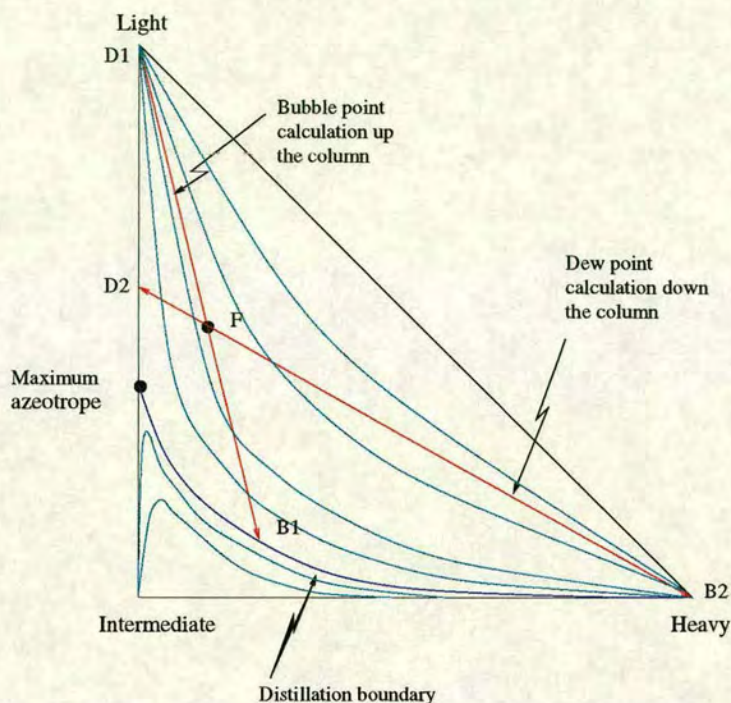


Figure 6.6: Direction of stagewise calculations

A *major* limitation of this model is its inability to provide solutions to a large class of ternary separation problems - those where the binary azeotrope and the entrainer are the terminal points of the distillation lines, as shown in figure 6.7. For these cases, the separations required to obtain solution flowsheets are not described by a distillation line. Hence, the model cannot obtain the separation outputs since they cannot be reached during operation at total reflux. The model does provide solutions to those problems where the entrainer introduces a distillation boundary such that the pure azeotrope components lie in different regions and are terminal points of the distillation lines in those regions. Using the total reflux model this nonideal synthesis work can attempt solutions to such systems. However, while the targetting and recycles (described in the following section) are not subject to the same limitations, a more general nonideal separation design model is needed to fully exploit them.

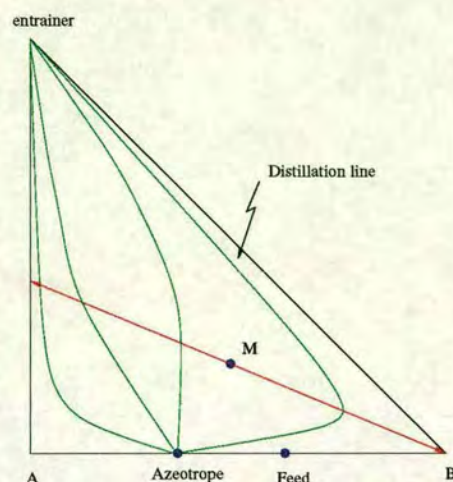


Figure 6.7: Example where total reflux operation is infeasible

6.4 Introduction of Recycles

Azeotropic separation problems require recycles for two reasons. Firstly, if an entrainer is added to facilitate separation then recovery and reuse of that entrainer makes economic sense. Second, processes which use nonsharp separations to produce pure component product sets usually require internal recycles (Wahnschafft et al. [44]). For nonideal separation problems, it is possible to end up with the desired product set but be left with streams which cannot undergo further separation by distillation, i.e. the azeotropes. To obtain complete separation of all streams and to maximise component recoveries in the products a recycle is required.

The approach taken for the inclusion of recycles in nonideal separation synthesis involves the identification of possible recycle species by the user before synthesis. For nonideal separation problems this is possible. The recycle species encountered in solutions to these problems are rich in the entrainer used and in the azeotropes which exist in the given system of components. Given that the recycle species are identified *a priori*, the most straightforward means of producing solution flowsheets is to introduce a flow of the recycle species into the flowsheet and accept as solutions processes which produce the desired product set *and* recover the recycle species as separation outputs.

Recycles have not hitherto been used in this work. Their inclusion requires that the

recycle alternatives and the specifications used to identify recovered recycle species are defined. This work is described separately for the entrainer and azeotrope recycles since they were found to behave in different ways. Following this, the integration of the recycles into the MCP synthesis method occurs through the addition of new structure generation operations to manage the recycle alternatives.

6.4.1 Entrainer recycles

The alternatives for the entrainer recycle are largely under the control of the designer. He/she is responsible for the provision of the entrainer species and for setting up the entrainer flow alternatives. For this work, only one species is permitted for a given problem. However, another synthesis problem could be created using a different species. The available entrainer alternatives for synthesis and the effect specification used to identify recovered entrainer recycle streams are described.

Synthesis alternatives

The alternatives considered for the entrainer recycle are the flowrate of the entrainer and the point of addition of the entrainer to the process flowsheet. The former has a very large influence on the economics of the process. In the example (figure 6.8) provided, a small entrainer flowrate leads to nonsharp separations in the solution flowsheet but more such separations are required. A larger flowrate results in more material being processed by each separator and sharper separations. However, the solution flowsheet contains less separators.

The flowrate itself is not used as the design variable. The bounds on its value which produce the full range of mixing with any process stream are not known in advance. Instead, the percentage distance along the process stream-entrainer mixing line is used. Its bounds are 0% and 99%: the former results in a zero flow for the entrainer and the latter in a mixture with a composition very close to the pure entrainer. 100% is not used since it is approached asymptotically. The value of the percentage mixing

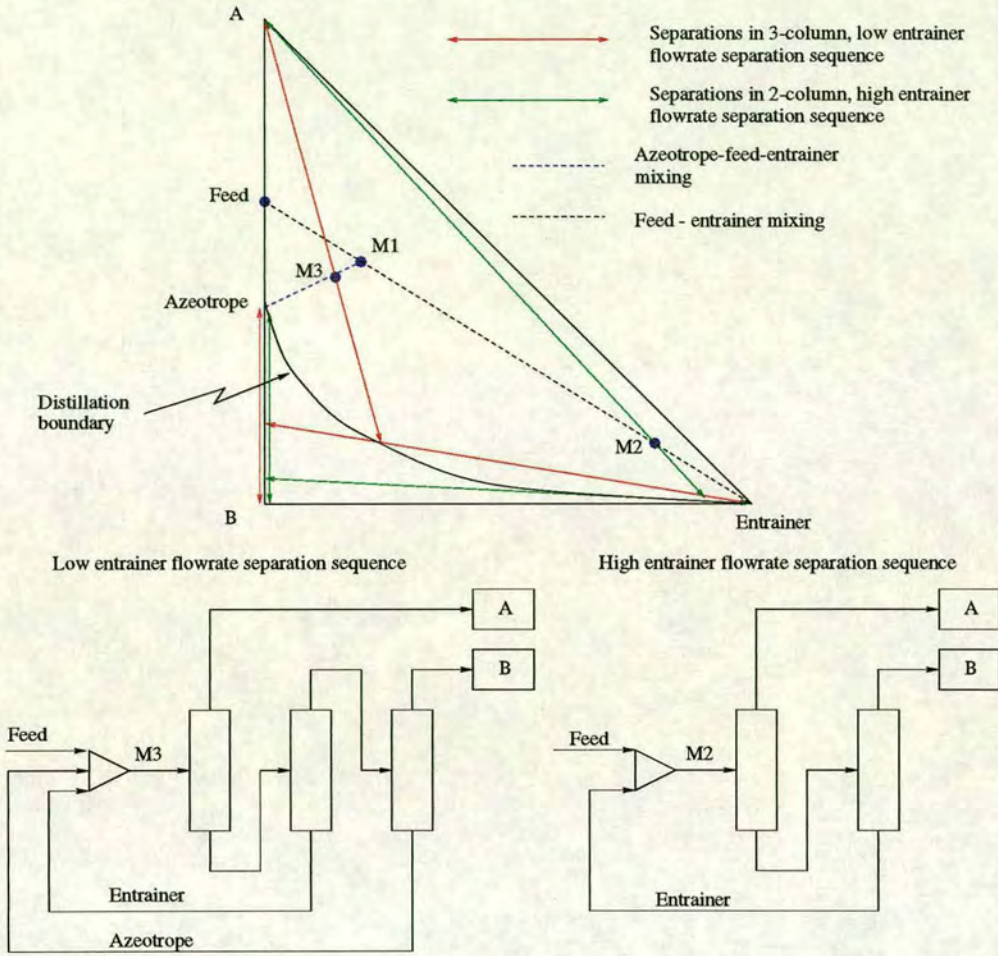


Figure 6.8: Influence of magnitude of entrainer flowrate on separation processes

is assigned by design variable discretisation (section 3.4) given these bounds and the number of design levels and distribution desired by the designer. The entrainer flow, which is not assigned until the stream with which the entrainer is mixed is known, is given by equation 6.1. SF is the flow of the process stream and PM is the value of the percentage mixing.

$$Entrainer\ flow = SF \left[\frac{\frac{PM}{100}}{1 - \frac{PM}{100}} \right] \quad (6.1)$$

The addition of the entrainer to the process flowsheet is considered for each process stream front encountered. The entrainer can mix with one or none of the IG streams created by the last separator unit added (analogous to the internal blending operation

described in section 3.3.3). This does not lead to as many alternatives as one might suppose since processes without entrainers quickly lead to distillation fixed points, i.e. pure components and azeotropes, which cannot undergo further separation. The entrainer must mix with a process stream before or at this stage if it is to be of use in generating feasible separation sequences.

Entrainer specifications

During synthesis, the introduced entrainer stream will blend with some process stream (F) to produce a mixed stream (M) which will undergo separation to form the products $P1$ and $P2$. Wahnschafft [44] has defined a *mixing goal* which requires that the recovered recycle stream preserves that separation task. This mixing goal is described by a set of equations which in turn define a region in composition space.

$$x_f + s(x_r - x_f) = x_1 + t(x_2 - x_1) \quad (6.2)$$

where x_f is a process stream composition, x_1 is the composition of product $P1$, x_2 is the composition of product stream $P2$ and x_r is the composition of the trial recycle stream. If this equation set has a solution where the parameters s and t are between 0.0 and 1.0 the trial recycle stream has satisfied the mixing goal. Figure 6.9 shows the mixing goal in operation for the acetone-benzene-chloroform system where benzene acts as an entrainer. The recycle composition defined by the shaded region preserves the separation by keeping the feed to the separator on the $P1$ - $P2$ line. This is an effect product specification - it defines a region of compositions whose function is to preserve a separation task.

The application of the mixing goal specification during synthesis requires that the outputs of the separation of the entrainer-process stream mixture are recorded. These and each trial entrainer recycle form the inputs needed to test for the satisfaction of the mixing goal. It does not become a recycle target until all these inputs have been provided. At the moment, the synthesis method assumes that a stream can satisfy only one target. If more than one target is satisfied by a stream, the synthesis method cannot choose between them. The region defined by the mixing goal can have one of the

azeotrope components on its boundary. In order to avoid having a potential product being accepted as an entrainer recycle, product targets are given priority over recycle targets, i.e. a stream is tested for product satisfaction before recycle satisfaction. To reinforce this a composition specification is added such that the recovered entrainer stream is within an interval box defined about the introduced entrainer composition. The size of this interval box is at the designer's discretion. However, it should be used to eliminate the overlap between the entrainer target and pure component product specifications.

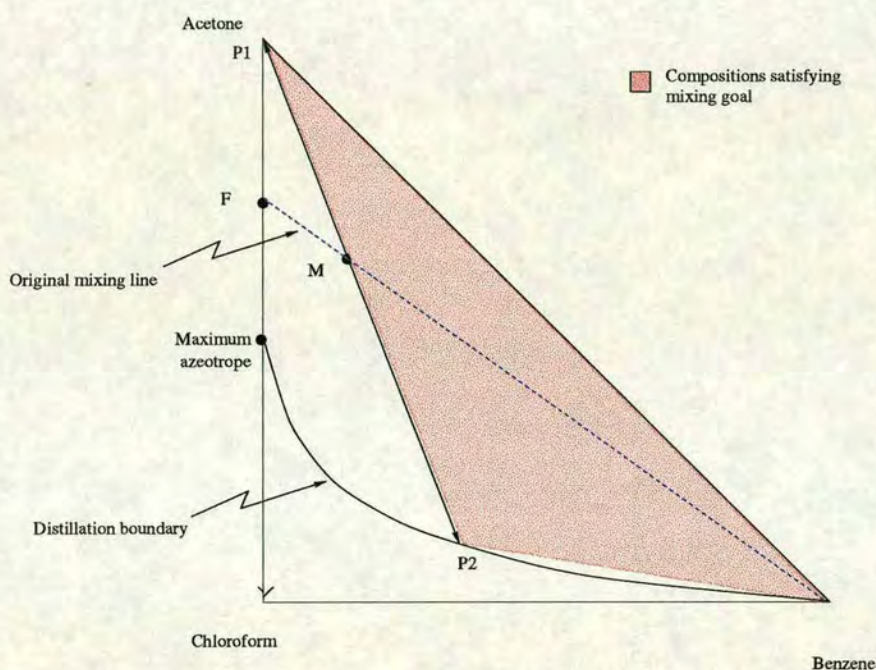


Figure 6.9: The mixing goal preserves the separation task

6.4.2 Azeotrope recycles

Azeotrope recycles exhibit different behaviour to entrainer recycles. In the first attempt at its inclusion, the azeotrope recycle was treated in the same manner as the entrainer recycle: discretisation of the percentage mixing variable determined the recycle flow and the introduced recycle stream composition was set to the azeotrope composition. However, this approach had difficulty in locating feasible solutions. An analysis shows that feasible azeotrope recycles are constrained and yields an effect specification for these recycles. This analysis is described, followed by the alternatives for azeotrope

recycles it produces and the effect specification used to identify recovered azeotrope recycles during synthesis.

Analysis of azeotrope recycles

The flows of the entrainer and azeotrope recycles are coupled, i.e. the amount of entrainer added to a process determines the flow of the azeotrope recycle in a solution flowsheet. The link between the two recycles is best shown through an analysis of the component recoveries of a separation limited by the presence of an azeotrope or a distillation boundary.

The analysis starts by looking at the maximum recovery of components obtainable in a binary azeotropic separation. Figure 6.10 illustrates the separation problem.

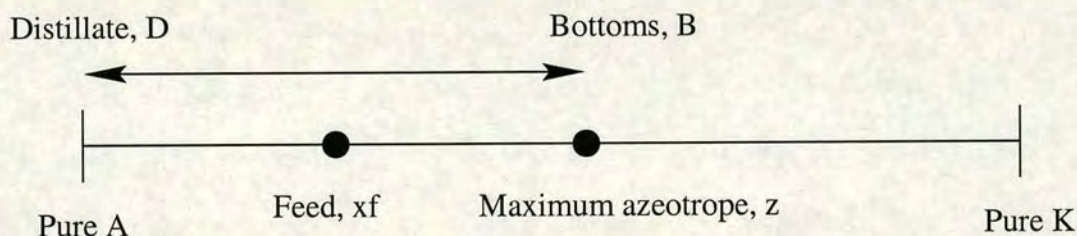


Figure 6.10: Binary azeotropic separation taken to its limits

The aim is to determine the effect of the azeotropic composition on the recovery of the component which can be obtained as a pure product, i.e. component A for the example. The component mass balance for A is:

$$Fx_{fa} = Dx_{da} + Bx_{ba} \quad (6.3)$$

If the separation is taken to its maximum extent then the distillate composition is pure A, i.e. $x_{da} = 1.0$, and all of the other component K is bound to the azeotrope output.

$$Fx_{fa} = D + Bz_a \quad (6.4)$$

$$Fx_{fk} = Bz_k \quad (6.5)$$

Equation 6.5 is rearranged such that the bottoms flow, B , is expressed in terms of the other variables and then substituted into equation 6.4 to give:

$$D = F \left[x_{fa} - x_{fk} \left(\frac{z_a}{z_k} \right) \right] \quad (6.6)$$

where the distillate is pure A. The equation is put on a more general basis by altering the labelling such that the pc subscript indicates the component which can be obtained pure and the az subscript indicates the component which is completely bound to the azeotrope. The flow of the pure component product, P_{pc} , is:

$$P_{pc} = F \left[x_{fpc} - x_{faz} \left(\frac{z_{pc}}{z_{az}} \right) \right] \quad (6.7)$$

The effect of the introduction of a recycle flow, R , of composition x_r which is blended with the feed stream, is now examined. Firstly, a mixture, M , of composition x_m , is obtained.

$$Mx_m = Fx_f + Rx_r \quad (6.8)$$

The pure component product flow after maximum separation is obtained from equation 6.7 with the mixture stream substituted for the feed stream.

$$P_{pc} = M \left[x_{mpc} - x_{maz} \left(\frac{z_{pc}}{z_{az}} \right) \right] \quad (6.9)$$

By expressing M , x_{mpc} and x_{maz} in terms in terms of the feed and recycle streams, the feed and recycle stream contributions to the product flow can be determined.

$$P_{pc} = F \left[x_{fpc} - x_{faz} \left(\frac{z_{pc}}{z_{az}} \right) \right] + R \left[x_{rpc} - x_{raz} \left(\frac{z_{pc}}{z_{az}} \right) \right] \quad (6.10)$$

The important conclusions drawn from equation 6.10 are:

- *The purpose of the recycle is to contribute to component recoveries.* To do so, the recycle composition must be chosen such that:

$$\frac{x_{rpc}}{x_{raz}} > \frac{z_{pc}}{z_{az}} \quad (6.11)$$

This constrains the recycle compositions and shows that a recycle composition equal to that of the limit of separation contributes nothing to the component recovery.

- The addition of an entrainer serves to alter the limit of separation and hence the ratio of $\frac{z_{pc}}{z_{az}}$. It can eliminate the limit of separation resulting in a zero flow azeotrope recycle for any recycle composition or, if the limit still exists, increase the range of recycle compositions which satisfy constraint 6.11 as illustrated by figure 6.11. This is an important result as it reveals more about the two classes of feasible entrainer described in section 5.5. Entrainers which introduce a distillation boundary retain the limit of separation and therefore require azeotrope recycles for complete separation. The other entrainer class, which keeps the feed azeotrope components in the same region, does not need azeotrope recycles.

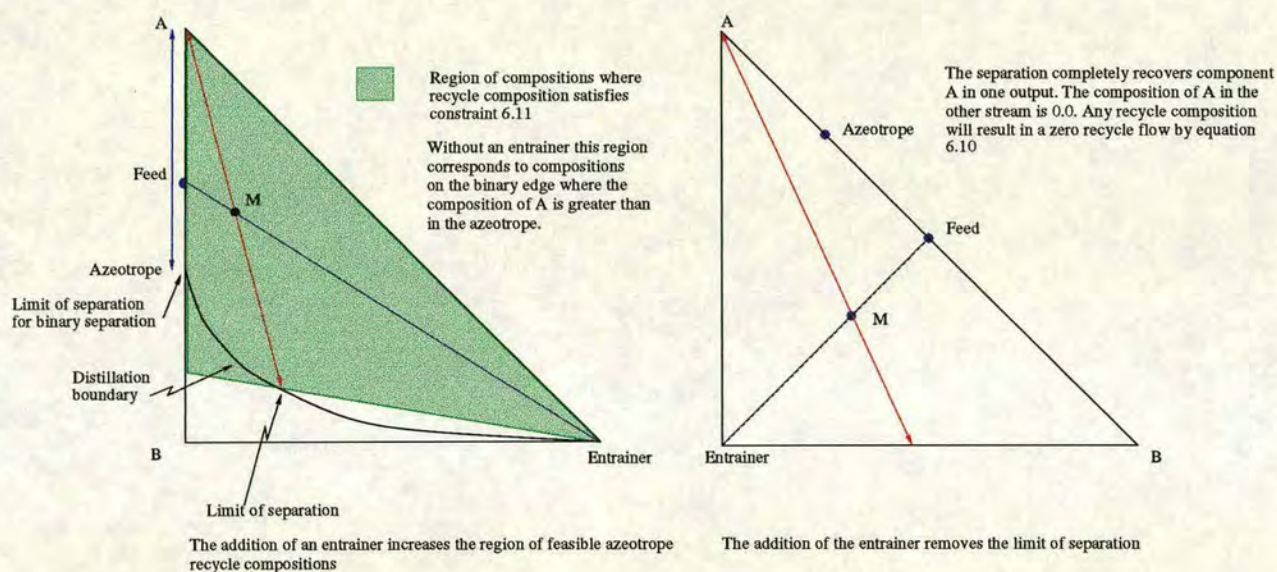


Figure 6.11: The limit of separation and the azeotrope recycle

Azeotrope recycle alternatives

The approach to recycles for this work is to introduce a flow of the recycle species into a process and then require that solution flowsheets produce the desired product set and recover the recycle species as separation outputs. The previous analysis is used to set the flow and composition of the introduced azeotrope recycle stream such that it contributes to the recovery of the azeotropic components. Equation 6.10 is employed to determine the azeotrope recycle flow by rearranging it into the following form:

$$R = \frac{P_{pc} - F \left[x_{fpc} - x_{faz} \left(\frac{z_{pc}}{z_{az}} \right) \right]}{x_{rpc} - x_{raz} \left(\frac{z_{pc}}{z_{az}} \right)} \quad (6.12)$$

F and x_f are the flow and composition respectively, of the stream, SE , whose separation is enhanced by first mixing with the azeotrope recycle, i.e. it produces more of the pure component pc . z is the limit of that separation and it sets up the region of feasible azeotrope recycle compositions from which an initial guess for the recycle composition, x_r , is chosen. Finally, a target for the recovery of the pure component, P_{pc} is set - it is the flow of that component in the stream SE , i.e. full recovery of the component from that stream is desired. Once the values of these variables are fixed the recycle flow, R , can be determined.

The synthesis alternatives correspond to the different values available to the inputs described. Since the choice of recycle flow and composition are dependent on the limit of separation, the azeotrope recycle alternatives correspond to the different limits which can be encountered for a given flow of entrainer. Three cases exist for the separation of a binary azeotrope:

1. The limit of separation after the entrainer has mixed with the separation problem feed.
2. The limit of separation after the entrainer has mixed with the azeotrope.
3. The binary azeotrope itself.

The limit of separation for entrainer-feed mixing is illustrated in figure 6.12. A feasibility requirement is attached to feed-entrainer mixing from the convex side of a distillation boundary. If this mixing does not result in a mixture on the concave side then it is declared infeasible since only one of the azeotrope components can be obtained as a distillation product. For separations limited by a concave distillation boundary, the azeotrope composition always satisfies the contribution constraint given by equation 6.11. This is chosen as the recycle composition, while the separation which produces

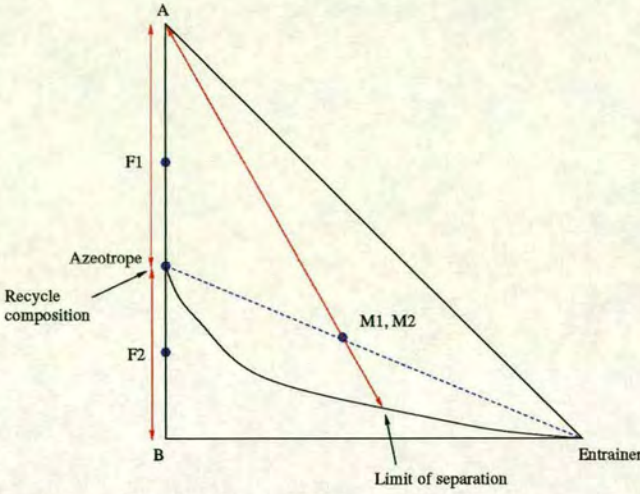


Figure 6.13: Limit of separation for entrainer-azeotrope mixing

azeotrope. Of course, if the feed to the separation process is the azeotrope then this alternative is not considered.

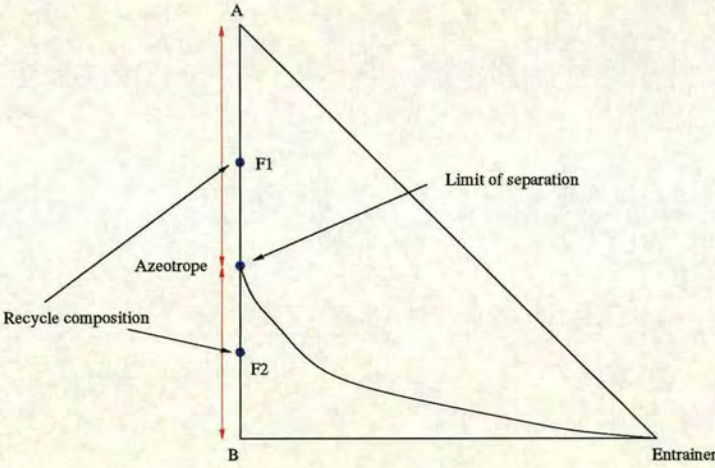


Figure 6.14: Limit of separation is the binary azeotrope

For the cases where the process stream - entrainer mixture is used to determine the limit of separation, that limit must be maintained. The azeotrope stream which is designed to maximise a component recovery from that mixture will blend with it, alter the mixture composition and hence the limit of separation. As a result, the component recovery can be reduced. To compensate for this the entrainer flowrate is increased such that the original limit of separation is preserved. Figure 6.15 illustrates the effect of the azeotrope mixing and the adjustment to the entrainer flowrate.

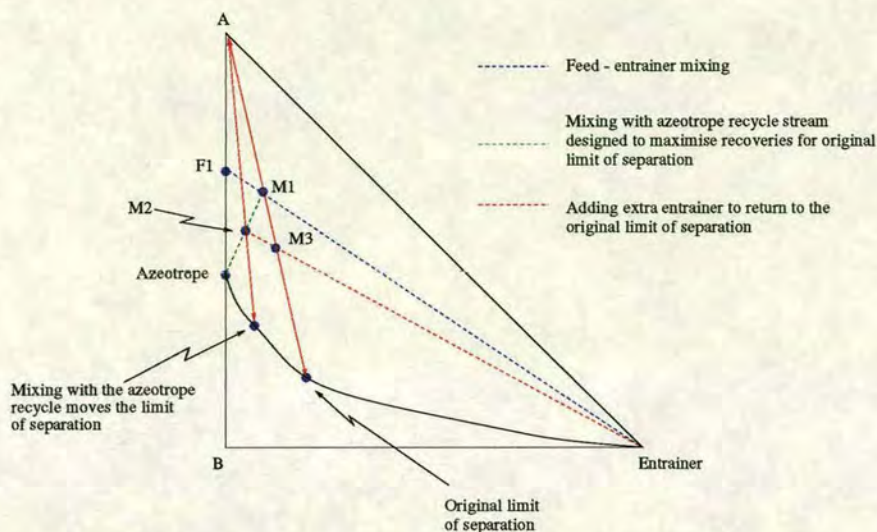


Figure 6.15: Adjusting entrainer flowrate to preserve the limit of separation

Each azeotrope recycle alternative is designed to contribute to a component recovery in a particular separation. Therefore, it makes sense to ensure that the recycle is introduced at the point where it performs this function. The alternatives considered for the addition of the azeotrope recycle to the process are the same as for the entrainer, i.e. for each process stream front encountered, the azeotrope recycle can mix with one or none of the IG streams created by the last separator unit added. However, the blending which corresponds to each limit of separation is specified and incorporated into the synthesis problem state - when the calculated azeotrope recycle flowrate is greater than zero the inputs to the blender which will include the azeotrope recycle are recorded, eg. for case 1 they are the process feed, the entrainer and the azeotrope. Blending alternatives which do not match the required input set are then rejected.

Azeotrope recycle specifications

The condition required for an azeotrope recycle to contribute to component recoveries is known. It is the constraint described by equation 6.11 and illustrated in figure 6.11. This *contribution constraint* forms an effect specification which ensures that azeotrope recycles perform their function. However, the area defined by the constraint, while characterising the azeotrope recycle, can overlap with potential pure component products and the mixing goal region. Conflicts with the former are resolved by giving

pure component targets priority over the recycle targets. The conflict with entrainer mixing goal is overcome by placing an interval box about the introduced azeotrope recycle composition within which the recovered recycle stream must lie. Its purpose is to eliminate the overlap between the two recycle targets and its size is controlled by the designer.

6.4.3 Enumerating the recycle alternatives

The recycle alternatives have been defined. For the entrainer, they are the percentage mixing values created by discretisation of that design variable and the point of addition of the entrainer to the flowsheet. For the azeotrope they are the recycles which arise from the different limits of separation that can be encountered and the resulting point of addition of the azeotrope stream to the process flowsheet.

The MCP synthesis method uses the structure generation operations (section 3.3.3) to control the appearance of different units in the process and manage the associated structural and design alternatives. Three new such operations are defined for recycles and are integrated into the cycle of structure generation operations. *Recycle creation* outputs the recycle streams which will be involved in the process. This operation is carried out once, at the very start of synthesis. The *recycle split* operation encodes the decision for a recycle stream to blend with the current process stream front. Finally, *recycle blending* controls the blending of recycle streams with the available process streams. The last two combined control the point of addition of each recycle stream to the process flowsheet. In order to select the inputs to these operations three new stream types were defined in a manner analogous to the existing process stream types. The *recycle newstream* (RS-NS) is a newly created recycle stream. This is split into two fractions: the recycle loiter-with-intent stream (RS-LWI) and recycle immediate gratification stream (RS-IG). The latter stream type is available for blending with a process stream, whereas the former bypasses that operation. A description of each operation follows using the breakdown of the operation function, its input streams, failure cases, structure created and the structural alternatives available. The integration of each of the new operations into the cycle of structure generation operations is then outlined.

Recycle creation

This operation is responsible for the introduced recycle streams alternatives. The first alternative fixed is the value of the entrainer mixing percentage. This is then one of the inputs to the procedure which enumerates the alternatives for the the azeotrope recycle.

Input streams	The operation has no input streams.
Failure cases	There are no failure cases. This operation is always successful.
Structure created	The structure created is a set of recycle newstreams, one for each recycle which can occur in the process.
Structural alternatives	The operation possesses only a single structural alternative, the set of output streams corresponding to the set of possible recycles.

Recycle split

This encodes the decision for a recycle stream to blend with the current process stream front. The intention is that the entire recycle stream flow will blend with a process stream, i.e. no fragmentation of the recycle stream is allowed. Therefore, the splitters which this operation introduces are only allowed to take on values of 0.0 and 1.0 for the split fraction.

Input streams	The input streams to this operation are recycle newstreams and recycle LWI streams.
Failure cases	There are no failure cases. If there are no inputs to the operation then a null operation occurs, i.e. no units or streams are created resulting in no change to the synthesis problem.
Structure created	A splitter is created for each input stream. The output from each splitter is a recycle LWI - recycle IG pair. The splitter design variable is its RS-IG split fraction
Structural alternatives	The operation possesses a single structural alternative, a splitter for each input stream.

Recycle blending

This operation controls the blending of recycle IG streams with the current process stream front. It is analogous to the internal blending operation described in section 3.3.2 and gives each recycle stream state the opportunity to blend with each new intensive stream state created by the feed set and CT units in the flowsheet. However, an important difference is that the blenders introduced by this operation have the process *newstreams* created by the previous internal blending operation as their inputs. It was felt that an IG-LWI operation between the two did not provide useful alternatives and therefore needlessly increased the search space. On a process flowsheet, this effectively combines the two blending operations. As mentioned in the recycle split operation, the entire recycle stream flow is to blend with a single process stream. Therefore, the splitter which distributes the recycle stream between the available blenders is only allowed to take on the values of 0.0 or 1.0 for the split fraction value, i.e. its function is to choose between the possible destinations for the recycle stream.

Input streams	Input streams are the recycle IG streams and process <i>newstreams</i> created by previous internal blending operation.
Failure cases	For cases where the azeotrope recycle IG stream is involved, the operation is failed when none of the input sets to the blenders matches the input set specified by the current azeotrope recycle alternative.
Structure created	Instances of blenders correspond to the number of process newstreams. The recycle IG streams are distributed among these using splitters.
Structural alternatives	The operation possesses a single structural alternative.

The modified structure generation algorithm

Figure 6.16 shows how the recycle operations are included in the cycle of structure generation operations. Recycle creation occurs only once and starts the algorithm by introducing the recycle streams which will be involved in the process. The first IG-LWI operation sets the inputs for the following internal blending operation, while the recycle split controls the inputs to the subsequent recycle blending operation. These similar operations are paired up. The position of the recycle split and recycle blending operations allows the recycle streams to first have the opportunity to blend with the process feed set and subsequently with the stream states created by the central transformation units. Figure 6.16 shows the structure created by the recycle operations and their interaction with the process streams.

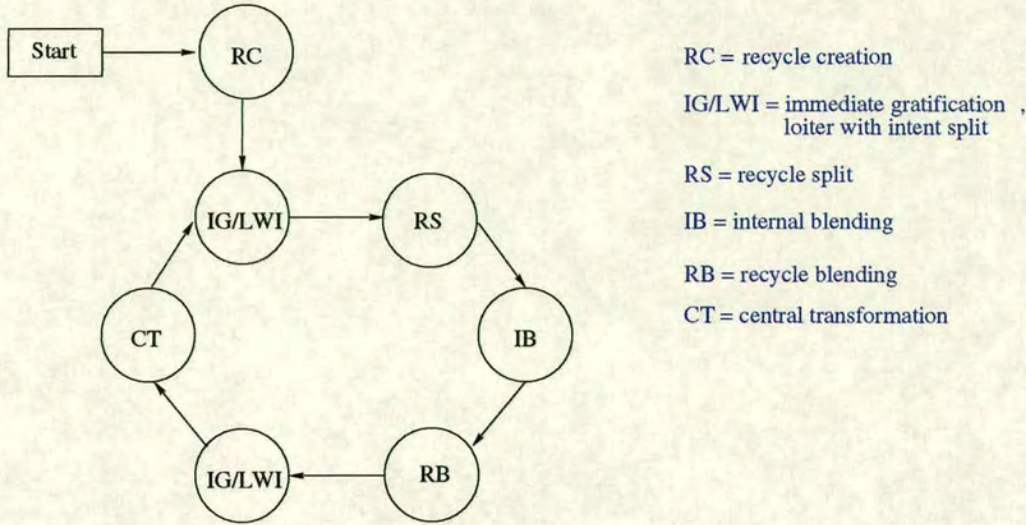


Figure 6.16: Including recycle operations in cycle of structure generation operations

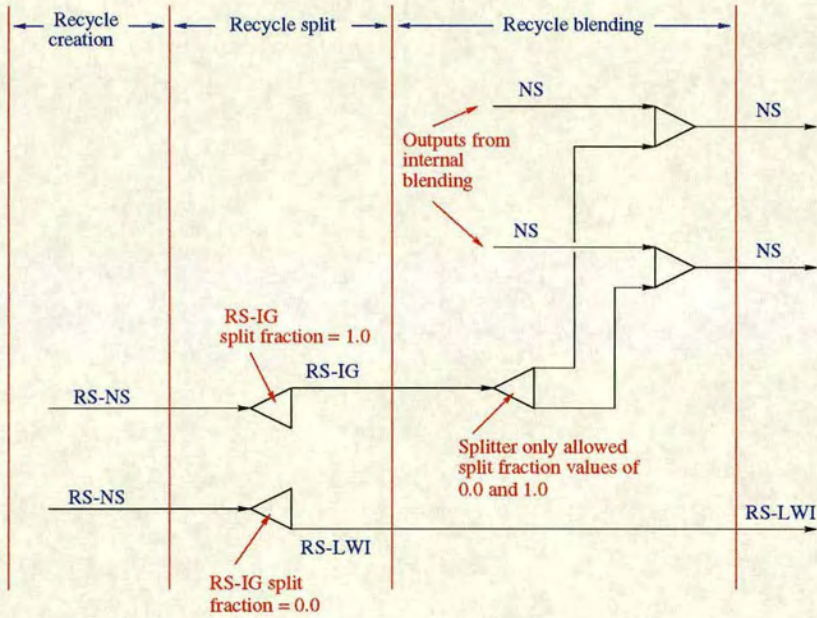


Figure 6.17: Structure created by recycle operations

6.5 Summary and conclusions

The steps taken to equip the MCP synthesis method with the means to solve nonideal separations are the targetting of feasible separations, the development of a shortcut separation design model and the inclusion of recycle alternatives. Targetting relies on user input to construct distillation line maps but thereafter provides for the identification and specification of feasible separations during synthesis. For systems where total reflux operation is feasible, the shortcut nonideal distillation design model returns quick estimates for column design, outputs and costs.

The approach to the inclusion of recycles identifies the recycle species in advance, introduces them into the flowsheet and then seeks to recover them by separation. Entrainers enhance separations through elimination of the limit of separation which allows the azeotrope components to be separated, or by moving the limit of separation such that the feasible region for the separation of the azeotrope components is expanded. Azeotrope recycles are required to contribute to the recovery of the azeotrope components. An analysis of these recycles delivered the condition which satisfies this requirement and also demonstrated that azeotrope and entrainer recycles are coupled. Thus, while the entrainer alternatives are largely under the control of the designer, azeotrope recycle alternatives are dictated by the system of components, the entrainer flowrate and the different limits of separation which these lead to.

The associated recycle targets are fuzzy - the recovered recycle stream does not have to be of the same state as the introduced stream. Instead, a pair of effect product specifications define a region of compositions for the recovered streams. The mixing goal for entrainers preserves the separation which the introduced entrainer and process stream mixture undergoes. The contribution constraint ensures that azeotrope recycles increase the recovery of the azeotrope components in the final product set. The region of compositions available to each recycle creates opportunities for the use of nonsharp separations for the recovery of the recycle species during synthesis. The modified MCP synthesis method is now applied to two nonideal binary separation problems.

Chapter 7

Nonideal Separation Synthesis Results

7.1 Introduction

The MCP synthesis method, equipped with nonideal system information, the targetting of feasible separations and the ability to handle recycles is applied to two nonideal binary separation problems. The first involves the separation of a binary maximum azeotrope using a heavy entrainer. The second problem uses a light-boiling entrainer to separate a minimum boiling binary azeotrope. Both systems form distillation boundaries and therefore each requires an azeotrope and an entrainer recycle to completely recover the feed components. The results will show that the synthesis method is capable of generating the known sets of solutions for different feed compositions. It has also led to the discovery of a new solution structure.

7.2 Acetone chloroform separation

Acetone and chloroform give rise to a maximum boiling azeotrope. Benzene is the entrainer used to separate them. The resulting ternary system has a distillation boundary which places the azeotropic components in two different distillation regions. Figure 7.1 shows the residue curve map generated for these components by the ASPEN PLUS simulator using the Wilson method to model the liquid phase nonidealities. Two different feed compositions are to be separated - the first lies on the concave side of the distilla-

tion boundary, the second on the convex side. With the exception of the feed streams, the remaining inputs are the same for each problem - the product specifications, thermodynamics, column model settings, recycle specifications, splitting and blending and search settings.

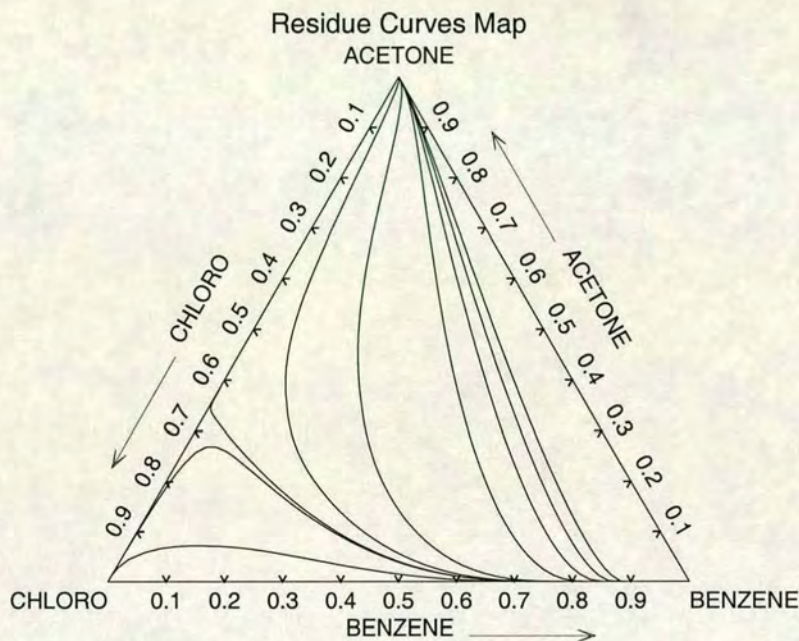


Figure 7.1: Acetone, chloroform, benzene residue curve map

Product specifications

The pure component products are specified in terms of the component recovery and composition. The lower bound on the former is low but will require a higher recovery than is possible from a single binary separation of the feed. Since the introduced and recovered recycle streams are not required to exactly match, this may lead to component recoveries slightly greater than 1.0 in the final product set. To avoid missing solutions where this occurs the recovery is given an arbitrarily high value, i.e. the upper bound is effectively removed. The lower bound of 0.95 on the product composition is intended to accept as solutions separations which, with some fine-tuning, are capable of producing a pure component product. Tables 7.1 and 7.2 contain the product specifications.

Specification	Lower bound	Upper bound
Fractional recovery	0.8	10.0
Composition	0.95	1.0

Table 7.1: Acetone product specifications

Specification	Lower bound	Upper bound
Fractional recovery	0.8	10.0
Composition	0.95	1.0

Table 7.2: Chloroform product specifications

Thermodynamic and physical properties

The Wilson method was chosen to model the nonideal interactions in the liquid phase. The binary interactions between the components were obtained from the ASPEN PLUS databank. Table 7.3 shows the interaction data for this system.

	Acetone	Chloroform	Benzene
Acetone	1.0000	1.0712	1.1198
Chloroform	1.6814	1.0000	3.3682
Benzene	0.5393	0.2793	1.0000

Table 7.3: Wilson parameters for acetone-chloroform-benzene system

Azeotropic column model settings

The only design variable not set by the targetting procedure is the column pressure. It is given a single discrete level of atmospheric pressure (1.01325bar) and 3 units are made available for synthesis.

Recycle specifications

The mixing goal and the contribution constraint are the main specifications for the entrainer and azeotrope recycles respectively. Satisfaction of these effect specifications fulfills the basic requirements for a feasible nonideal separation process. Each recycle also has a flow and composition specification which reduces the regions defined by the effect specifications in order to avoid target satisfaction conflicts. Acceptable flows for

both recycles are within 25% of the introduced flow. Streams which have up to a 25% difference in mole fraction value of the dominant component in the introduced recycle are accepted.

Splitting and blending settings

The IG-LWI split fraction is allowed to take on values of zero and one. The internal blending operation handles the blending of process streams while the recycle operations control the blending of recycle streams with process streams.

Search settings

For this work, all solutions returned are of interest and therefore a large solution set size (100) is used. For the given inputs, this set will not be filled, resulting in an exhaustive search of the synthesis tree. A plant life of 1 year is specified and the same utilities and costings as used for the multicomponent product problems are made available here. The cooling utility is water at 288K and the heating utility is steam at 14bar.

7.2.1 Feed on concave side of distillation boundary

An equimolar feed stream is used. Its flow is 100kmol/hr and its temperature and pressure are set to ambient levels. Table 7.4 contains the relevant data. The entrainer recycle mixing design variable is set to have a lower bound of 20% and an upper bound of 70%. Two discrete levels result in the values at the bounds being used as the design alternatives.

Flow (kmol/hr)	100
Temperature (K)	298
Pressure (bar)	1.01325
Composition (mole fraction)	
Acetone	0.5
Chloroform	0.5

Table 7.4: Acetone-chloroform separation: concave feed data

The synthesis method returns 12 solutions to the input problem in just over 30 minutes. The unit design time comprises 10% of this total, the design variable targetting 83% and the search the remainder. Table 7.5 shows the costs of the solutions obtained, omitting the repeated designs (same structure, column designs and cost). The flowsheets of the best structurally different solutions are shown in figures 7.2 - 7.6. Solutions 10,11 and 12 perform unnecessary and costly blending, eg. mixing a pure component product with the recovered entrainer, or add a column to purify an almost pure product stream. Diagrams of these solutions are not shown.

Solution	Cost (millions of UK pounds)	Relative cost
1	2.1	1.00
2	2.5	1.25
3	2.6	1.25
5	2.9	1.39
7	3.0	1.46
8	3.4	1.63
9	3.4	1.66
10	3.6	1.76
11	3.7	1.76
12	3.8	1.81

Table 7.5: Costs of solutions to concave binary feed separation

The best two solutions returned have a recycle structure not yet encountered in solutions to nonideal binary separation problems. Figure 7.2 illustrates solution 1: solution 2 has the same structure but uses a higher entrainer flowrate. Referring to figure 7.2, a flow of azeotrope-rich recycle equal in composition to the feed is introduced into the first column and recovered in the second. The entrainer is introduced into the second column and recovered in the third. This solution beats the best of the conventional solutions (solution 3) by about 400,000 UK pounds. The bulk of the cost saving occurs in the column which recovers the benzene entrainer; the final column in solution 1 and the second column in solution 3. The separation between chloroform and benzene has a long total reflux path resulting in a large number of stages for this separation in both solutions (158 for solution 1 and 164 for solution 3). However, solution 1 does not carry the azeotrope recycle flow into this separation and also uses a lower entrainer flow. The resulting vapour rate is 70kmol/hr compared to 259kmol/hr for the chloroform-benzene separation in solution 3. The difference in cost between the respective columns is over

half a million UK pounds and this is reflected in the total solution costs.

Solution 7 is the known two-column solution which uses a large entrainer flow to take advantage of the closeness of the distillation boundary to the binary edge as it approaches the entrainer fixed point. While this reduces the number of columns needed to obtain the product set, the high mass flowrate through the columns increases both the operating and capital costs. The low concentration of chloroform in the feed to the near-binary separation results in a high minimum reflux and hence total reflux ratio which again adds to the column costs. However, given the shape of the distillation boundary for this system, i.e. it closely approaches the chloroform-benzene edge at low benzene concentrations, a reduction in the entrainer flowrate and therefore in costs may be possible without greatly reducing the purity of the final product set.

Solution 8 has a distinctly different separation structure to the other solutions. Both the outputs from the first semi-sharp separation are feeds to different columns. This solution works because the high entrainer flowrate generates binary outputs from that first separation. Again, as with solution 7, this high flowrate carries a large cost. Also, the high concentration of acetone in the feed to the acetone-chloroform separation results in a high recovery of that component in the product stream. However, the recovery can never be complete since a fraction of acetone is bound to the bottoms product of that separation. The latter column output flow is absorbed by the entrainer recycle. If the entrainer flowrate is reduced, a semi-sharp separation of the entrainer-feed mixture would lead to a reduction in acetone recovery as more of this component would be bound to the distillation boundary output from the first separation in the process. An azeotrope recycle would then be required to completely recover the feed components.

Solution 9 first separates the binary feed to recover a fraction of the acetone in the top product and the azeotrope in the other. Thereafter, the entrainer is mixed with the azeotrope stream and follows the same processing path as solution 7.

7.2.2 Feed on convex side of distillation boundary

For this problem the feed composition lies on the opposite side of the distillation boundary to that of the first. Its flow is again 100kmol/hr and its temperature and pressure are set to ambient levels. Table 7.6 contains the relevant data. The entrainer recycle mixing design variable is set to have a lower bound of 20% and an upper bound of 70%. Three discrete levels distributed uniformly between the bounds will result in two design alternatives at the bounds and one centered between them.

Flow (kmol/hr)	100
Temperature (K)	298
Pressure (bar)	1.01325
Composition (mole fraction)	
Acetone	0.1
Chloroform	0.9

Table 7.6: Acetone-chloroform separation: convex feed stream data

The synthesis method returns 20 solutions in 37 minutes. The unit design time comprises 9% of the total, the unit design targetting 83% and the search the remainder. Again, repeats of solutions are obtained and the separation targetting or column solution method are suspected. Table 7.7 shows the costs of the solutions obtained, omitting the repeated designs.

Solution	Cost (millions of UK Pounds)	Relative cost
1	2.4	1.00
3	2.5	1.02
5	2.8	1.14
7	2.9	1.20
9	3.3	1.37
11	3.3	1.37
15	3.3	1.39
16	3.4	1.42
18	3.5	1.45
19	3.5	1.46

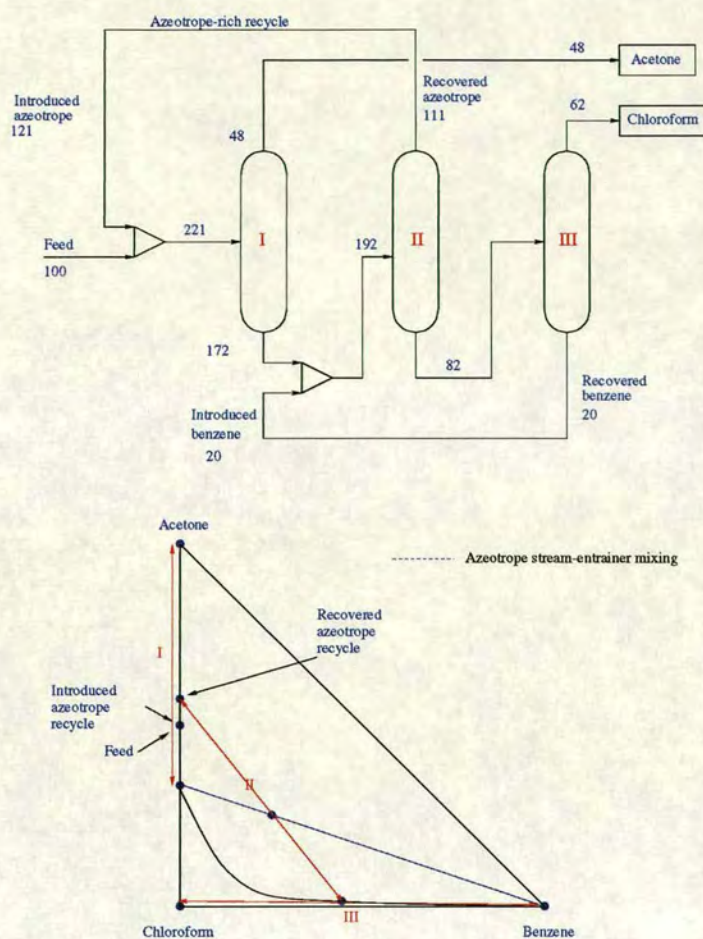
Table 7.7: Costs of solutions to convex binary feed separation

Solutions 1 and 18 are of most interest as these are particular to convex feeds. Their flowsheets and column design information are given in figures 7.7 and 7.8 respectively. The remaining solutions are similar to those obtained for concave feed compositions: the

distillation boundary is first crossed using entrainer blending such that the separation process starts in the concave region.

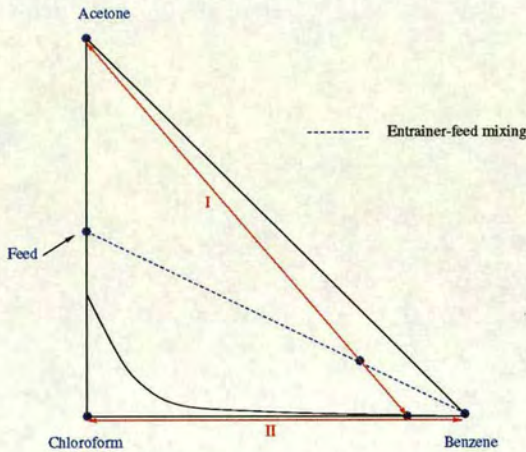
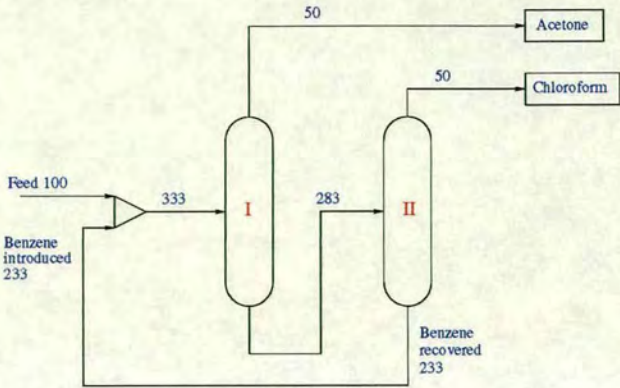
Solution 1 is a known solution and is analogous to the best solution obtained for the concave feed as the azeotrope-rich recycle is introduced to the separation limited by the azeotrope composition. However, unlike the concave feed solution, no staggering of the recycles to reduce the load on the chloroform-benzene separation is possible. The production of an azeotrope recycle stream from the second column richer in chloroform than the limit of separation for the first column (i.e. the azeotrope) requires a boundary-cross. This is not a feasible option since the feed to column II is not inside the boundary-crossing region.

Solution 18 is analogous to solution 9 for the concave problem. Again, the high entrainer flowrate results in a high mass flow through the chloroform-benzene separation causing it to dominate the solution costs.



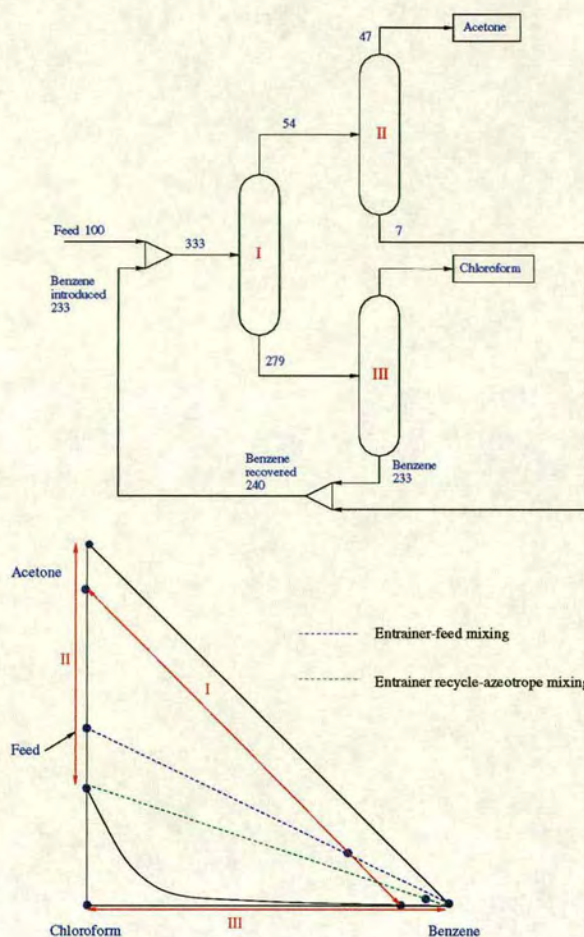
Variable	Column I	Column II	Column III
Light key	Acetone	Chloroform	Chloroform
Heavy key	Chloroform	Benzene	Benzene
Light recovery in tops	0.44	0.45	1.0
Heavy key recovery in bottoms	1.0	1.0	1.0
Operating pressure (bar)	1.01	1.01	1.01
Distillate flow rate (kmol/hr)	48	110	61
Minimum reflux ratio	9.0	1.0	1.0
Operating reflux ratio	10.9	1.6	1.1
Minimum no. of stages	23	14	51
Operating no. of stages	46	42	158
Condenser duty (kW)	400	923	520
Reboiler duty (kW)	1442	706	180
Annual operating cost (million UKP)	0.25	0.20	0.08
Capital cost (million UKP)	0.57	0.49	0.6

Figure 7.2: Concave feed acetone-chloroform separation - Solution 1



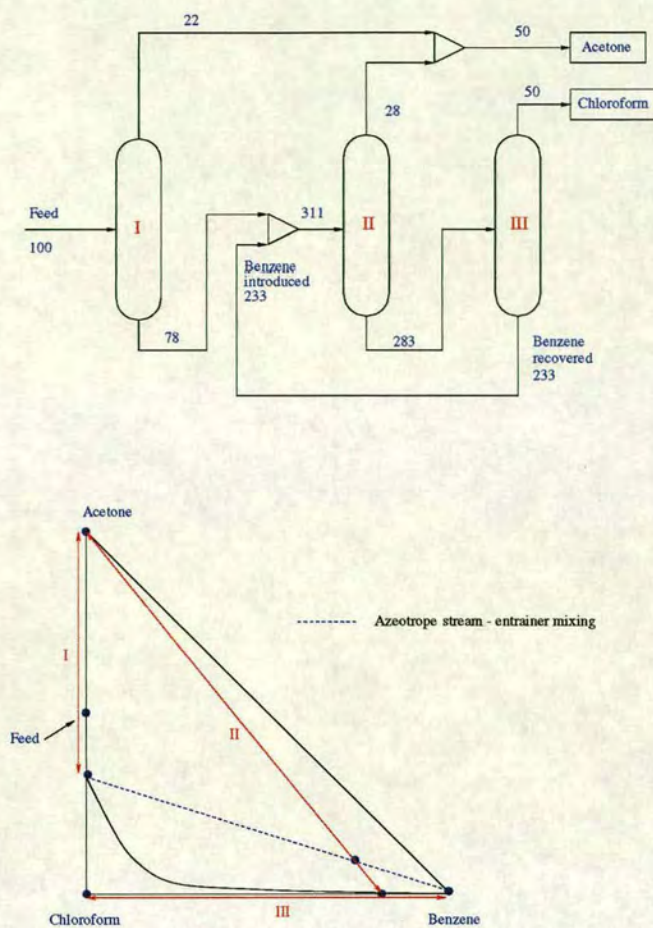
Variable	Column I	Column II
Light key	Acetone	Chloroform
Heavy key	Chloroform	Benzene
LK tops recovery	0.99	1.0
HK bottoms recovery	1.0	1.0
Operating pressure (bar)	1.01	1.01
Distillate flowrate (kmol/hr)	50.0	50.1
Minimum reflux ratio	4.2	25.9
Operating reflux ratio	5.1	31.1
Minimum reflux ratio	29	60
Actual reflux ratio	59	117
Condenser duty (kW)	413	422
Reboiler duty (kW)	2517	2093
Annual operating cost (million UKP)	0.47	0.34
Capital cost (million UKP)	0.49	1.9

Figure 7.4: Concave feed acetone-chloroform separation - Solution 7



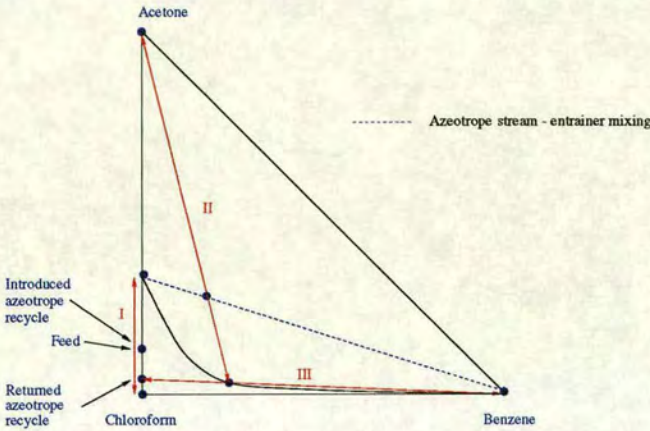
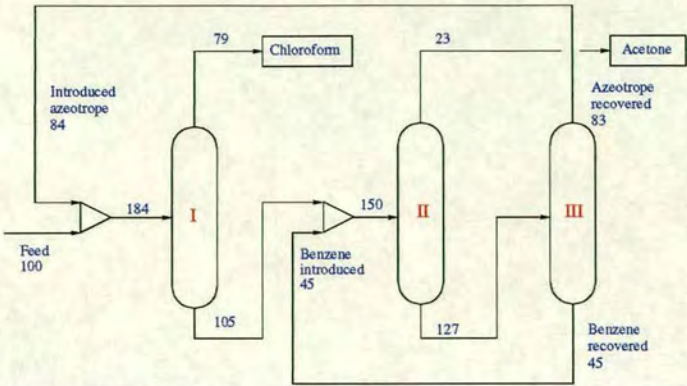
Variable	Column I	Column II	Column III
Light key	Acetone	Acetone	Chloroform
Heavy key	Chloroform	Chloroform	Benzene
LK tops recovery	0.99	0.95	1.0
HK bottoms recovery	0.91	1.0	1.0
Operating pressure (bar)	1.01	1.01	1.01
Distillate flow (kmol/hr)	54.6	47.3	49.1
Minimum reflux ratio	3.7	1.1	29.4
Operating reflux ratio	4.4	1.4	35.4
Minimum number of stages	24	25	60
Actual number of stages	50	76	116
Condenser duty (kW)	454	391	384
Reboiler duty (kW)	2477	61	2093
Annual operating cost (UKP)	0.40	0.05	0.34
Capital cost (UKP)	0.45	0.33	1.80

Figure 7.5: Concave feed acetone-chloroform separation - Solution 8



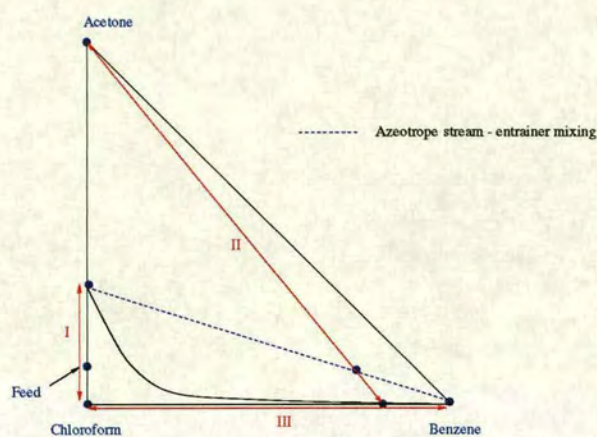
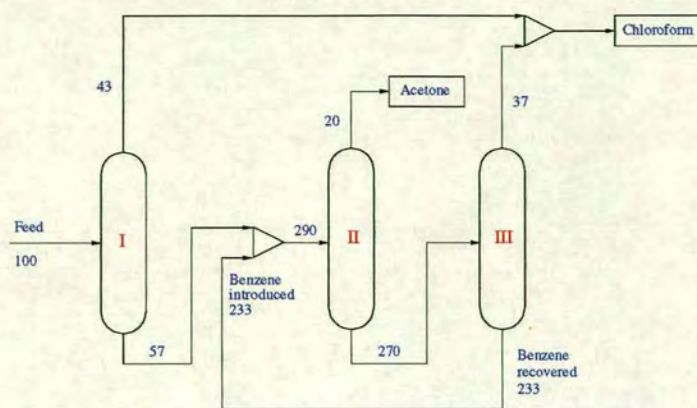
Variable	Column I	Column II	Column III
Light key	Acetone	Acetone	Chloroform
Heavy key	Chloroform	Chloroform	Benzene
LK tops recovery	0.44	0.99	1.0
HK bottoms recovery	1.0	1.0	1.0
Operating pressure (bar)	1.01	1.01	1.01
Distillate flow (kmol/hr)	21.9	28.1	50.1
Minimum reflux ratio	9.1	6.8	25.9
Operating reflux ratio	10.9	8.1	31.1
Minimum number of stages	23	28	60
Actual number of stages	46	56	117
Condenser duty (kW)	181	232	422
Reboiler duty (kW)	653	2516	2093
Annual operating cost (million UKP)	0.11	0.38	0.34
Capital cost (million UKP)	0.37	0.44	1.78

Figure 7.6: Concave feed acetone-chloroform separation - Solution 9



Variable	Column I	Column II	Column III
Light key	Chloroform	Acetone	Chloroform
Heavy key	Acetone	Chloroform	Benzene
LK tops recovery	0.54	0.61	1.0
HK bottoms recovery	1.0	1.0	1.0
Operating pressure (bar)	1.01	1.01	1.01
Distillate flowrate (kmol/hr)	79.0	22.6	82.5
Minimum reflux ratio	4.7	6.0	2.2
Operating reflux ratio	5.7	7.2	2.6
Minimum number of stages	27	30	61
Actual number of stages	55	60	165
Condenser duty (kW)	664	187	693
Reboiler duty (kW)	879	1104	404
Annual operating cost (million UKP)	0.2	0.18	0.14
Capital cost (million UKP)	0.61	0.38	0.90

Figure 7.7: Convex feed acetone-chloroform separation - Solution 1



Variable	Column I	Column II	Column III
Light key	Chloroform	Acetone	Chloroform
Heavy key	Acetone	Chloroform	Benzene
LK tops recovery	0.54	0.99	1.0
HK bottoms recovery	1.0	1.0	1.0
Operating pressure (bar)	1.01	1.01	1.01
Distillate flow (kmol/hr)	42.9	20.0	37.2
Minimum reflux ratio	4.7	8.4	38.4
Operating reflux ratio	5.7	10.1	46.1
Minimum number of stages	27	25	60
Actual number of stages	55	50	116
Condenser duty (kW)	361	166	314
Reboiler duty (kW)	477	2407	2092
Annual operating cost (million UKP)	0.11	0.36	0.33
Capital cost (million UKP)	0.44	0.39	1.86

Figure 7.8: Convex feed acetone-chloroform separation - Solution 18

7.3 Toluene Isopropyl-alcohol separation

Toluene and isopropyl-alcohol (IPA) give rise to a minimum boiling azeotrope. Acetone is the entrainer used to separate them. The resulting ternary system has a distillation boundary which places the azeotropic components in two different distillation regions. Figure 7.9 shows the residue curve map generated for these components by ASPEN PLUS. Two different feed compositions are to be separated - the first lies on the concave side of the distillation boundary, the second on the convex side.

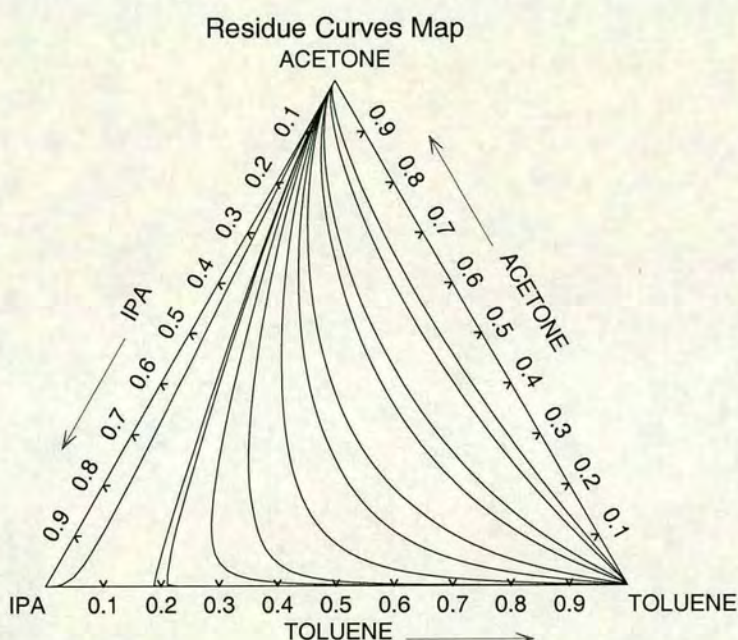


Figure 7.9: Residue curve map for acetone,IPA, toluene system

Only the component-component interaction data for the problem changes. All other inputs remain the same as for the previous problem. Table 7.8 shows the interaction data for this system.

	IPA	Toluene	Acetone
IPA	1.0000	0.1785	0.9086
Toluene	0.5037	1.0000	0.8505
Acetone	0.5343	0.6586	1.0000

Table 7.8: Wilson parameters for Acetone-IPA-toluene system

7.3.1 Feed on concave side of distillation boundary

An equimolar feed stream is used. Its flow is 100kmol/hr and its temperature and pressure are set to ambient levels. Table 7.9 contains the relevant data. The entrainer recycle mixing design variable is set to have a lower bound of 20% and an upper bound of 80%. Three discrete levels result in the values at the bounds and one value centered between them.

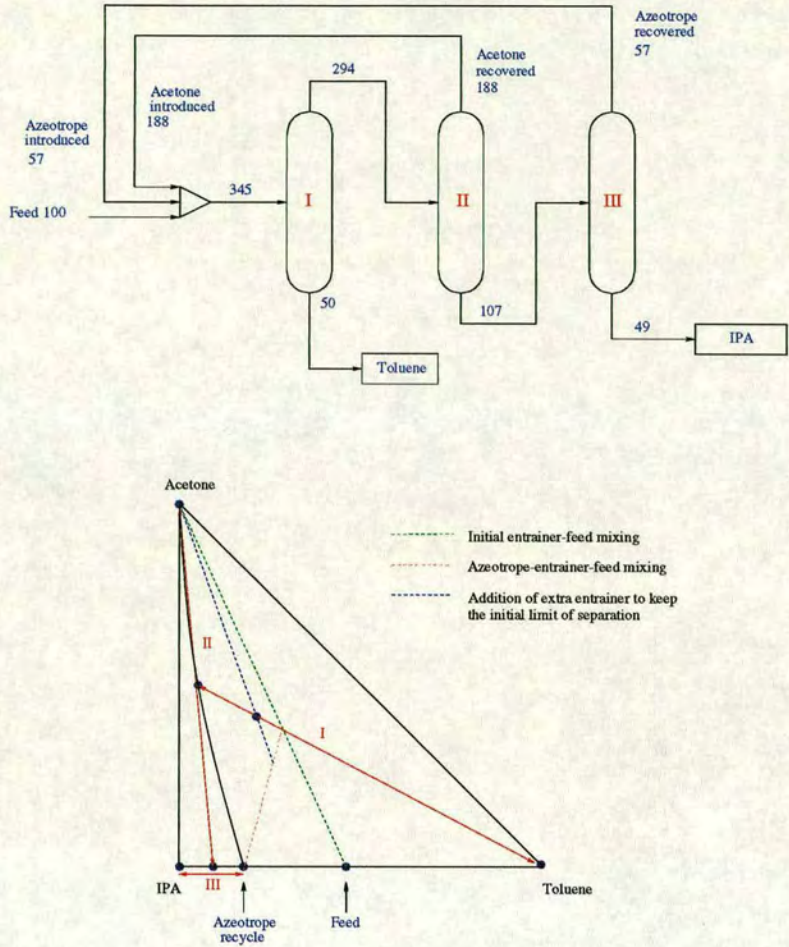
Flow (kmol/hr)	100
Temperature (K)	298
Pressure (bar)	1.01325
Composition (mole fraction)	
IPA	0.5
Toluene	0.5

Table 7.9: IPA-toluene separation: concave feed stream data

The synthesis method returns 12 solutions to the input problem in just under 30 minutes. The unit design time comprises 35% of this total, the design variable targeting 44% and the search the remainder. The costs of the solutions obtained are shown in table 7.10, omitting the repeated solutions.

Solution	Cost	Relative cost
1	2.2	1.0
3	2.4	1.1
5	2.6	1.2
7	3.1	1.4
9	3.3	1.5
11	4.1	1.8

Table 7.10: Costs of solutions to concave binary feed separation



Variable	Column I	Column II	Column III
Light key	IPA	Acetone	Toluene
Heavy key	Toluene	IPA	IPA
LK tops recovery	1.0	1.0	1.0
HK bottoms recovery	0.80	1.0	0.52
Operating pressure (bar)	1.01	1.01	1.01
Distillate flowrate (kmol/hr)	294	107	57
Minimum reflux ratio	0.5	1.86	2.08
Operating reflux ratio	0.6	2.23	2.50
Minimum number of stages	10	25	17
Actual number of stages	39	70	48
Condenser duty (kW)	2553	1553	530
Reboiler duty (kW)	498	975	444
Annual operating cost (UKP)	0.15	0.31	0.12
Capital cost (UKP)	0.51	0.77	0.36

Figure 7.10: Concave feed IPA-toluene separation - Solution 1

Variable	Column I	Column II	Column III
Light key	IPA	Acetone	Acetone
Heavy key	Toluene	IPA	IPA
LK tops recovery	1.0	1.0	1.0
HK bottoms recovery	0.7	0.6	1.0
Operating pressure (bar)	1.01	1.01	1.01
Distillate flow (kmol/hr)	91	108	90
Minimum reflux ratio	0.5	0.5	1.83
Operating reflux ratio	0.6	0.6	2.19
Minimum no. of stages	8	9	24
Operating no. of stages	31	35	68
Condenser duty (kW)	843	920	660
Reboiler duty (kW)	488	591	254
Operating cost (million UKP)	0.26	0.29	0.47
Capital cost (million UKP)	0.16	0.13	0.11

Table 7.11: Acetone-Toluene-IPA concave feed separation - new solution structure column designs

The 12 solutions returned possess the same structure - the conventional three column solution, containing both an entrainer and an azeotrope recycle. The best of these is shown in figure 7.10. In contrast to the acetone-chloroform-benzene system, the distillation boundary does not approach a binary edge until it is close to the pure entrainer. As a result, the upper bound of 80% on the entrainer mixing does not create a mixture which can produce two pure components in a single separation. Increasing the entrainer mixing to 90% produces the conventional two-column solution costing 4.7 million UK pounds - the high entrainer flowrate of 900 kmol/hr contributing most to this cost.

In order to locate the new recycle structure solution, the recycle composition specifications and the bounds on the pure component recoveries were relaxed. For the former the composition interval about the introduced azeotrope recycle composition is increased to 50% and lower bound on the component recoveries is dropped to the same value. The flowsheet obtained is shown in figure 7.11 and the column designs are given in table 7.11.

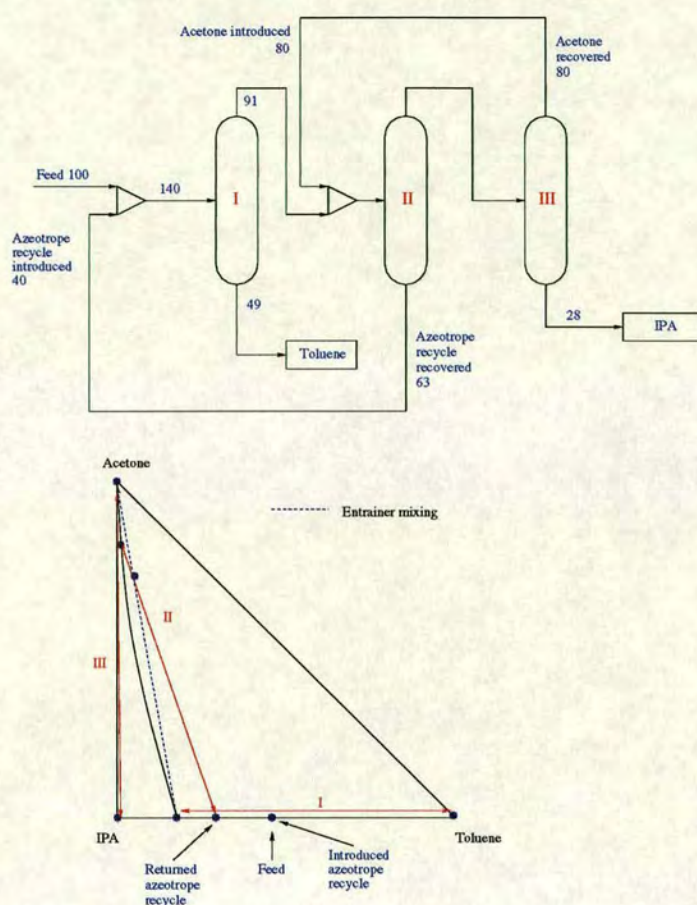


Figure 7.11: Acetone-IPA-Toluene concave feed separation - new solution structure

The returned solution structure satisfies both of the recycle effect specifications, i.e. the mixing goal for the entrainer recycle and the contribution constraint for the azeotrope recycle - the IPA recovery in the final product is 56% compared to 40% without the azeotrope recycle. The reason for the low IPA recovery is that the available separation alternatives do not enable the recovery of the azeotrope recycle at its introduced composition. The total cost of this solution is 1.4 million UK pounds. This is almost 800,000 UK pounds or 35% cheaper than the best solution identified so far. Though mass-balance satisfaction has yet to be obtained and the other solutions will need to be optimised, this remains a competitive solution structure for concave feed separation problems.

7.3.2 Feed on convex side of distillation boundary

The feed composition is now placed on the convex side of the distillation boundary. The feed flow remains at 100kmol/hr and the temperature and pressure are set to ambient levels. Table 7.11 contains the relevant data. The entrainer recycle mixing design variable is set to have a lower bound of 20% and an upper bound of 70%. Three discrete levels distributed uniformly between the bounds will result in two design alternatives at the bounds and one centered between them.

Flow (kmol/hr)	100
Temperature (K)	298
Pressure (bar)	1.01325
Composition (mole fraction)	
IPA	0.1
Toluene	0.9

Table 7.12: Feed stream data

A single solution is returned for the given inputs: the boundary is crossed using entrainer blending at the 70% mixing level and then follows the conventional 3-column concave feed separation with an azeotrope and entrainer recycle (see figure 7.10). The search time was 23.5 minutes of which the design took up 27%, the targetting 56% and the search itself the remainder. The cost of the solution is 3.5 million UK pounds.

The location of the solutions particular to convex feed separation problems, i.e. those which recover some or all of the IPA in the first separation in the process, was achieved by increasing the flow of the entrainer. A mixing level of 80% produced the solution which recovers all the IPA in the first separation. Its cost is 2.17 million pounds and its structure is shown in figure 7.12. The second convex feed-specific solution recovers IPA in the first and third separations and blends the two streams to give the final product. This required the entrainer flow to be raised to the 90% mixing level producing a solution costing 3.1 million UK pounds, illustrated in figure 7.13.

Complete recovery of IPA in the first separation provides cost savings in two ways. Firstly, it reduces the amount of entrainer required to achieve a given mixing % level in comparison to immediate blending of the entrainer with the feed. Second, the

introduced azeotrope-rich recycle is at the feed composition lowering the amount of separation which is required in the second and third columns in the process. The conventional 3-column and the second convex feed-specific solution require those separations to yield 3 fixed point products. Recovering all the IPA in the first separation requires that the second and third separations yield 2 fixed point products resulting in shorter separation paths, i.e. less separation and therefore lower costs.

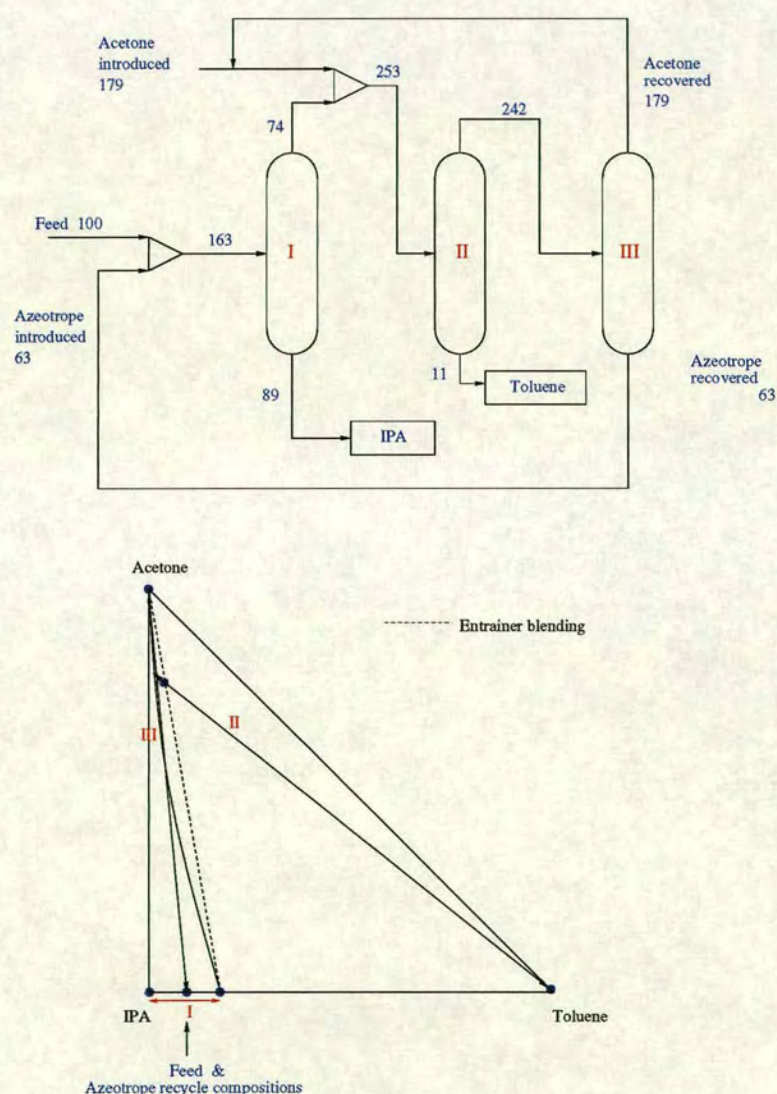


Figure 7.12: Acetone-toluene-IPA separation: convex feed-specific solution 1

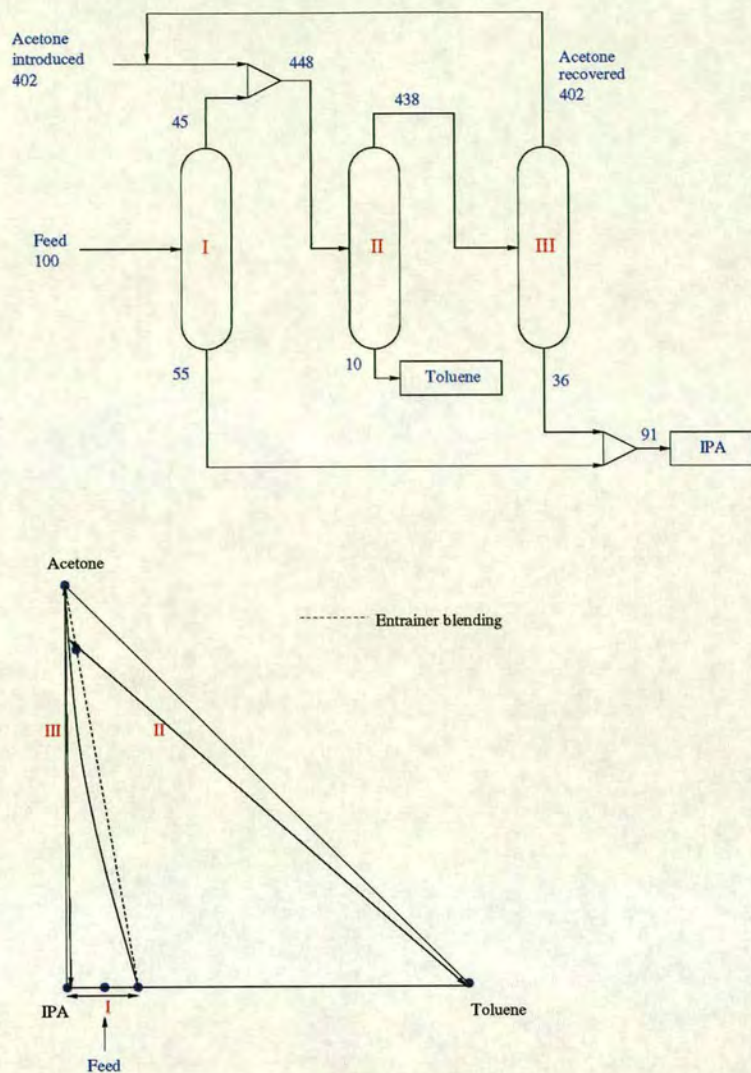


Figure 7.13: Acetone-toluene-IPA separation: convex feed-specific solution 2

7.4 Discussion of results

The results show that the extensions to the MCP synthesis method - the column model, targeting procedure and inclusion of recycles are capable of generating new and known solutions to the given nonideal ternary separation problems.

The straightforward problem setup enables the designer to experiment with different problem inputs for the location of solutions. In the acetone-chloroform-benzene system the initial problem inputs were sufficient to yield the solution set - the high degree of curvature of the distillation boundary allows the use of a wide range of entrainer

mixing levels in the solutions to both the concave and convex feed separation problems. In contrast, the low curvature of the distillation boundary in the acetone-IPA-toluene system resulted in the need to experiment with the problem inputs in order to locate a range of solution structures - different runs were carried out using different levels of entrainer mixing and product and recycle specifications were relaxed demonstrating that the synthesis method is most effective when the designer makes good use of his/her controls over the search space and solution acceptance.

The availability of a set of solutions gives the designer the means to develop an understanding of the given problems. For this work, the results obtained show that solutions requiring high entrainer flowrates are less competitive than those utilising lower entrainer flows - even if the separation can be carried out in 2 rather than in 3 columns. Economies of scale are violated here, i.e. a reduction in the number of processing units does not bring about a reduction in costs. The characteristics of the best solutions are lower flows through columns and shorter separation paths. For both ternary systems, regardless of feed position, these solutions are those in which the first separation completely recovers one of the feed components. For concave feed separation problems, this is the new solution structure discovered. While optimisation and more detailed designs and costings are required to fully determine solution rankings, these solutions are expected to remain among the best.

Chapter 8

Conclusions and Further Work

8.1 Introduction

There were two goals for the work described in this thesis:

1. The development of an preliminary design stage synthesis method for ideal and near-ideal multicomponent product problems.
2. To extend that method to ternary nonideal separation, pure component product problems.

Both these problems share two features. The first of these is the appearance of multicomponent products and in particular what have been termed *effect* products. These are not specified by composition but by a desired effect or property. Examples of such products appear in the petroleum industry where specifications on viscosity, density and vapour pressure etc., are common. For nonideal systems, effect specifications have been defined for the recycle streams required by feasible processes.

The second shared feature is the need for a greater exploration of stream space. For ideal and near-ideal systems, this is a direct result of the product specifications: these may occupy a region in stream space and satisfying them purely through sharp separation followed by blending is thermodynamically inefficient. In order to locate good

solutions composition regions other than those covered by these operations must be explored. Thus, opportunities are created for non-sharp separation as well as stream splitting and blending. For nonideal separation problems, the effect product specifications on the recycle streams combined with the limits on separation imposed by the presence of azeotropes and distillation boundaries also create a need for a greater exploration of stream space.

The interest in the early stages of process synthesis requires that the method explores a wide range of alternatives and returns a set of good solutions which can be then taken to the next, more rigorous stage of design, i.e. the method acts as a filter which excludes infeasible alternatives and the worst solutions. To be effective in this function, the method should automate, as far as possible, the generation and searching of the process alternatives. Finally, the synthesis method should be algorithmic as such methods are better suited to the generation of new processes. Heuristic or evolutionary methods rely heavily on knowledge of existing processes. Such knowledge is not available for MCP separation problems.

8.2 Multicomponent product synthesis

The MCP synthesis method developed to meet these requirements uses simple descriptions of a synthesis problem and unit function as its basis. The synthesis problem consists of the set of targets and the streams which do not yet satisfy them (the process stream front). Process unit function is described in terms of the contribution to intensive stream state diversity. The types of units made available are splitters, blenders and central transformation (CT) units. The latter are defined to be units which invest energy in a stream or set of streams to create a set of new intensive stream states. The operations which add the units to a flowsheet and hence alter the synthesis problem are organised so that they maximise the number of different intensive stream states created. The more different stream states created, the greater the exploration of stream space and the more possible paths to solution processes. The resulting algorithm for the generation of structural alternatives is a cycle of operations

which builds up the process structures given the feed stream set and the set of available central transformation units.

Unit design alternatives are provided through the use of design variable discretisation. For each unit design variable, upper and lower bounds on its value, the number of design levels and a distribution type are defined. These generate a set of alternative values for that variable. For a given unit, its set of design alternatives corresponds to all combinations of discrete values of its design variables. Design variable discretisation is problem simplification which has four important results:

- It reduces the number of alternatives to be considered.
- The alternatives can be enumerated during synthesis, i.e. in the same manner as the structural alternatives.
- Discretisation also puts the level of modelling rigour completely under the control of the designer: the thermodynamics, level of design detail and solution method are contained within the unit models. No limits are imposed by the synthesis method as it is only interested in the inputs and outputs from a unit design. It is not affected by the degree of non-linearity of the equations involved.
- Discretisation leads to a search space consisting entirely of discrete decisions, allowing that space to be cast in the form of a tree.

The size of the resulting search tree is constrained by placing a limit on the number of central transformation units which can appear in a flowsheet. Unlike sharp separation, pure component product problems, the addition of units does not necessarily bring a flowsheet closer to a solution. At the moment, very long or non-terminating paths cannot be identified in advance and a limit on the number of CT units seeks to prevent their occurrence. A deterministic, depth-first search with pruning is chosen to locate the best solutions to a given synthesis problem. The search algorithm sits on top of the structure generation algorithm and design variable discretisation and directs the order in which the synthesis alternatives are generated and therefore explored.

The results obtained for the MCP separation problems presented demonstrate that the method fulfills the aims of this work. Its ability to consider a range of separator types combined with the availability of stream splitting and blending opens up composition regions not explored by most of the existing synthesis methods. As a result new solutions are returned for the given literature problems containing only flash vessels and stream blending and a mixture of flash vessels and column separations. It has also been shown that the method can reproduce the stream bypassing and semi-sharp separation structures returned by the literature approaches. The oil stabilisation problem with its effect product specification, a large number of feed components and the requirement for nonsharp separators is tackled successfully by the synthesis method. A range of solution structures is obtained, containing flash vessels, compressors and stream blending. None of the literature approaches, except perhaps the NLP formulation of Smith and Pantelides [37] could attempt a solution to this problem.

The ability of the synthesis method to return sets of solutions and the straightforward problem setup encourage experimentation on the part of designer. The best performance in terms of the range of solutions returned is achieved when the designer makes good use of his/her controls over the search space and solution acceptance. In the case of the literature problems, recognising that exact satisfaction of the component recovery specifications is unlikely results in the relaxation of those specifications, allowing the location of a range of flash solutions. Subsequently, consideration of a limiting separation within those solutions leads to the addition of a semi-sharp separation to the set of available separations. The oil stabilisation problem benefits from a decomposition which results in better solutions to the vapour compression problem and hence the overall problem - without incurring a large increase in the overall search space. The final set of solutions obtained give a strong indication of the properties of good solutions to this problem. Thus, while the stated purpose of providing a set of solutions was to examine the effect of applying different costings and design considerations in later stages, it also helps the designer to understand what makes for a good solution under a single design consideration. The same cannot be said of methods which return only a single solution.

8.3 Nonideal separation synthesis

The extension of the MCP method to nonideal separation synthesis was undertaken with a view to exploiting, as far as possible, existing knowledge of nonideal systems. The work done was broken down into three sections: targetting of feasible separations, provision of a shortcut model for column designs and finally adding the ability to handle recycles.

Due to the limits of separation imposed by the presence of azeotropes and distillation boundaries design variable discretisation has difficulty in locating feasible separations. Therefore, those separations are targetted through the construction and analysis of distillation line maps. As the focus of this work is on synthesis rather than nonideal system properties, the approach taken relies heavily on user input to construct the maps but thereafter provides for the identification and specification of the feasible separation alternatives during synthesis. The set of alternatives is drawn from literature and their targetting mimics, using numerical procedures, the knowledge gained from the largely visual analysis of distillation line maps.

The next section of work consists of providing a shortcut model for nonideal separation column design. As with MCP synthesis the focus is on the early stages of the design process. Since a rigorous tray-by-tray design is at odds with this aim a shortcut model design is used. It is an existing model developed by Senos-Matias [34] and Herron [16] which is based on the Fenske-Underwood-Gilliland method. Nonideality is incorporated through the use of rigorous thermodynamics and a tray-by-tray infinite reflux calculation to determine the minimum number of stages. The contribution of this work is in making the solution method and the assignment of the design specifications robust enough to include the model in an automated synthesis method.

Recycles are required by nonideal separations to make the processes more economic and to maximise the recovery of the azeotropic components in the product streams. The former involves the recycling of an entrainer-rich stream and the latter recycles of azeotrope-rich streams. Since these recycles can be identified in advance, the simplest

means of synthesizing recycle flowsheets is to introduce a flow of a recycle stream into a process and then to accept as solutions flowsheets which recover both the pure components and the recycle streams as separation outputs.

Effect specifications are defined for the recycle streams. Entrainers use the mixing goal specification of Wahnschafft et al [44] which seeks to preserve the separation task on the entrainer-process stream mixture. This work defines a *contribution constraint* for azeotrope-rich recycles whose purpose is to ensure that these streams increase the recovery of azeotropic components in the final product set. This specification is derived using an analysis of the limit of separation imposed by an azeotrope. The same analysis also forms the basis for determining the available azeotrope recycle blending alternatives. The recycle alternatives are integrated into the MCP synthesis method through the addition of new operations to the existing set of structural generation operations.

The result is a preliminary design method for ternary nonideal separation problems. It retains the features of the underlying MCP synthesis method, i.e. the ability to return a set of solutions, a straightforward problem setup and simultaneous generation and searching through the process alternatives. Combined with extensions described the method provides a set of feasible recycle flowsheets complete with column designs. This reduces the designer's workload and provides the information to help him/her to develop an understanding of the solution space. For example, costly separations are easily identified either through the costs or column variables such as the number of stages, the reflux ratio and vapour rate. The magnitude of recycle flows required by the different solution structures can be compared and the impact on costs assessed. Finally, the location of new and known solutions to the given problems validates the approach taken and, in particular, the work on azeotrope-rich recycles.

8.4 Recommendations and Future Work

8.4.1 MCP Synthesis

It has been shown that the MCP synthesis method has met the requirements for an early design tool. Two main areas for future work have been identified.

The full set of stream bypassing and feed stream blending alternatives is not available for problems with more than 2 products and 2 feed streams. The method presented has wider applicability and explores more of composition space than its literature counterparts and the benefit is shown by its ability to generate new solutions. This is due in large part to the rich set of structural alternatives provided and it is therefore desirable to extend that richness to problems with higher numbers of feeds and products.

The MCP separation problem search space presents difficulties:

- Unlike pure component product separation problems the addition of units does not necessarily bring a flowsheet closer to a solution - long or non-terminating search tree paths are possible if the number of CT units are not limited.
- Increasing the number of discrete split fraction levels leads to an explosion in the breadth of the search space.

At the moment, pruning is the only means of reducing the search space for a fully defined problem. This cannot be relied upon to produce consistent reductions and it certainly does not aid in locating solutions. It is a technique whose effectiveness depends on how early in a search good solutions are located. If none exist or a solution set never reaches its specified size then the search is exhaustive. It is desirable to carry out the search in a more opportunistic manner such that the search space contains more solutions and that the better solutions are generated early in the search. This requires that a measure of which alternatives are closer to a solution than others be developed. This can then be used to determine which partial paths should be expanded first, hopefully speeding up the location of solutions for the benefit of pruning. It may

also be possible to use the measure to tune designs during synthesis such that the current flowsheet is closer to a solution.

8.4.2 Nonideal separation synthesis

The nonideal separation synthesis work has been used to generate new and known solutions to the given problems. It has the potential for use on a wider range of problems.

The column model is limited to those separations which can be modelled using a single total reflux path. As a result, it excludes an entire class of solutions to ternary nonideal separation problems - solutions which are within the capabilities of the targetting and recycle handling parts of method. Either the current shortcut design procedure must be adapted to overcome these limitations or a new method must be found.

Currently, the calculations involved in targetting separations can be applied to systems with more than three components *except* for the determination of the distillation boundary path. A general method for locating distillation boundaries is highly desirable, particularly if it can be applied during rather than before synthesis. This would lay the basis for attempting solutions to problems containing more than 3 components. It is also the key step in the construction of the distillation line map for a system during synthesis, further reducing the designer's workload.

Appendix A

Implementation of the MCP Synthesis Method

A.1 Introduction

This appendix provides an outline of the MCP synthesis method implementation, focussing on the organisation of the method information right up to the final tree search algorithm. The application of any process synthesis algorithm first requires that its inputs, the problem description and available process units, must first be constructed. The manner in which this is done is described in section A.2. Next, the objects which comprise the search tree and its solutions are described, followed by the structure generation operations. Finally, the search tree operator and solution handling procedures are outlined before presenting the search algorithm which makes use of them.

A.2 Structuring the synthesis problem information

Prior to the generation and searching of the flowsheet alternatives, the information which a synthesis method requires in order to perform these functions must first be assembled. This information forms the description or *state* of the synthesis problem. For this work the synthesis problem state is composed of four elements:

- The *target front* which contains the set of targets which must be satisfied by solutions.
- The set of streams which do not yet satisfy any of the targets - the *process stream front*.

- The *structure generation state* which provides the input for the cycle of operations responsible for controlling the appearance of process units.
- The *alternative enumeration state* which stores the information on the alternatives which have yet to be explored.

A.2.1 Synthesis problem hierarchy

Having defined the synthesis problem state the next stage is to structure the information which comprises it. Figure A.1 illustrates the hierarchy used to achieve this. A brief description of each element in the hierarchy is provided.

Implementating the hierarchy

All programming of the work undertaken was done using Fortran 90. As the work progressed it was necessary to take a more object-oriented approach to the implementation in order to manage the increasing complexity and to build a more robust application. While Fortran 90 does not support object orientation as languages such as C++ or Java are designed to do, it does not rule out working along such lines. Each object described is captured by a module. The module contains a definition of the object using a data structure (or structures). Various parameters or constants required by the object are defined. Specific instances of the object which are required for a synthesis problem, eg. a table of components, are declared. All functions associated with the object are held in the module. These tend to be grouped by their purpose, i.e. basic functions such as copying and information retrieval, presynthesis functions such as reading in problem data and putting it in the correct form and synthesis functions such as updating parts of the problem state. An object higher up in a hierarchy *uses* the module of an object lower down. In that way it has access to all the definitions and functions of the lower object.

Component

The description of chemical component is comprised of its name, formula and a set of properties such as molecular weight, normal boiling point, heat of formation etc., required for thermodynamics calculations. A table of chemical components is also defined and an instance of it declared to store the component table for a synthesis problem.

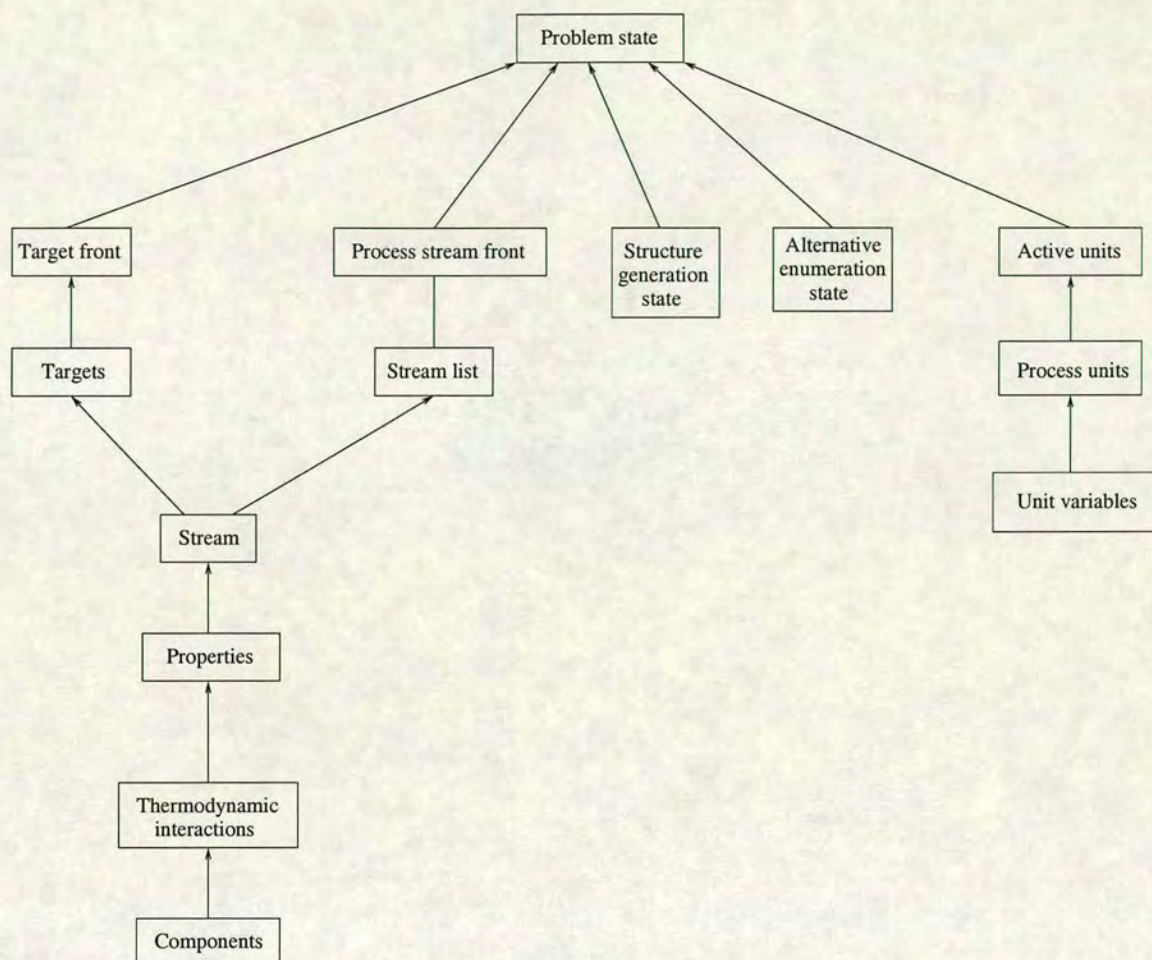


Figure A.1: Synthesis problem information hierarchy

Component-component interactions

This is a matrix of the component-component interaction parameters required for the evaluation of the nonideal thermodynamic properties. An instance of the interaction matrix is declared to store the parameters for a particular synthesis problem.

Thermodynamic properties

This consists of different sets of procedures for the evaluation of thermodynamic and physical properties of streams. Ideal vapour-liquid equilibrium (VLE) properties are based on the Kistiakowsky method [19] for hydrocarbon mixtures. Nonideal properties use the activity-based Wilson method [38]. Physical property estimation methods were obtained from Perry's Chemical Engineers' Handbook [31].

Stream

The definition of the stream state is important to this work. The state is comprised of two distinct parts:

- *Intensive stream state* describes the thermodynamic state of a stream, that part of the stream which is independent of flow. By Gibbs phase rule [38] $nc + 1$ independent variables are required. For example, temperature, pressure and $nc - 1$ mole fractions, where nc is the number of components in the stream.
- *Extensive stream state* is that part of the stream description which is dependent on the flow of a stream. Only a single variable need be used.

The stream object uses mole fractions to describe composition with temperature and pressure completing the intensive stream state description. The molar flow describes the extensive stream state. A parameter is set up before synthesis indicating which kind of thermodynamic properties are to be used (ideal or nonideal) by streams. Each stream has a pointer to the unit which created it and the CT unit which set up its intensive state.

Stream list

This contains the definition of a list of streams and the functions required to handle it, i.e. insertion, deletion etc..

Process stream front

The process stream front is one element of the problem state definition. It is the list of streams which do not yet satisfy the product specifications. Whenever streams are consumed or created by process units the process stream front is updated accordingly. Each stream in the PSF has a unique integer identification or id.

Target

Targets consist of a set of specifications, a value and a satisfaction state. The specifications are based on properties of the target entity itself, eg. a product stream is specified in terms of properties of a stream (flow, temperature, density etc.). A specification requires identification of the desired property, the reference temperature and pressure for the property evaluation (if needed), and the upper and lower bounds on its value. The target value is the value of the product stream expressed on a volumetric,

molar or mass basis. The satisfaction state of a target is indicated by a logical flag (true when satisfied, false otherwise). The id of the stream which satisfies the target is also recorded.

Target list

This contains the definition of a list of targets and the functions required to handle it.

Target front

This is another component of the problem state definition. The satisfaction state of the synthesis targets changes through synthesis. Therefore, targets must be associated with the current synthesis problem. The target front consists of a list of targets. Presynthesis functions are provided to read in all target information. Synthesis functions are used to update the target front for each new synthesis problem.

Unit variables

The unit variable object defines the variables required by process units. The definition includes the name of the variable (eg. temperature, pressure), the type of the variable (integer or real), the upper and lower bounds on the variable value and a slot for the value assigned to the variable. Discretisation information is also included and consists of an integer value which indicates the type of discretisation (uniform or logarithmic), the number of discrete levels and the current discrete level.

Process unit state

This object defines the process unit information required to describe the flowsheets constructed during synthesis. Flowsheets do not need a full description of a unit model, i.e. the equation set, variable set and the solution method used to design the unit. The type of unit, its fully defined variable set (i.e. all variables have been assigned values) and costs are sufficient. The former can be reused as inputs to the unit design if necessary. The variable set is broken down into three subsets:

- The *unit settings* are the group of variables which remain constant throughout synthesis, eg. the ratio of operating reflux to minimum reflux used in shortcut distillation column design.

- The *design variables* are those whose values must be set in order to solve the unit model equations.
- The variables whose values are fixed from the solution to the unit model equations make up the *calculated variables*.
- The unit cost, broken into capital and operating portions. This is kept separate from the calculated variable set as it is used to rank solutions to synthesis problems.

The process unit also has pointers to its input and output streams.

Active units

The information on the units available for synthesis is assembled and stored here, together with the functions for their retrieval during synthesis. Each unit type made available has an instance declared which contains all its variable information. The limit on each unit type's appearance in a flowsheet is also stored here.

Structure generation state

The information required to drive the cycle of structure generation operations is grouped together in the structure generation state. It includes:

- A pointer to the most recently designed CT unit, used primarily to enforce the mixing restriction in internal blending.
- The current CT unit type selected for inclusion in the process.
- A count of each CT unit type used in synthesis so far. This information is needed if a limit is imposed on the number of instances of CT units within a flowsheet.
- Information on the current structure generation operation - its position in the cycle of operations, the operation type and flags indicating if the operation is just starting or coming to an end. If the former is true the input set for the operation must be constructed. If the latter is true, then synthesis must progress to the next operation in the cycle.
- The inputs for the current alternative type within a given operation.

This module also stores the procedure which enforces the cycle of structure generation operations. For the given structure generation state this procedure works out the resulting operation and indicates whether or not an input set must be constructed for it.

Alternative enumeration state

This is used to keep track of the number of alternatives which are available to a problem. It consists of:

- Flags to indicate the first and last alternative. The former is used to initialise the information for alternative enumeration. The latter indicates that alternative enumeration is complete.
- The index to the current alternative.

Problem state

The problem state object contains slots for the target front, the process stream front, the structure generation state and the alternative enumeration state. The information contained within the active units object is made available to the synthesis problem but it is not part of the problem definition itself.

A.2.2 Unit models

The process unit state has already been defined. It consists of the unit type, its design variable set, the calculated variable set and the unit costs. This is all that is needed to define a unit within a flowsheet. However, the equation set for the unit must be solved using the specified design variables to determine the values of the calculated variables and costs and to fix the state of the output streams from the unit. Also, the unit state is a generic description of a unit - it contains no specific information about the different units available to synthesis. A model for each unit is required for synthesis. The function of the model is two-fold. Firstly, it must provide synthesis with all the necessary information about the unit. Second, it must store the unit equation set and provide the means for its solution.

The unit model consists of a set of procedures which perform the unit design and provide synthesis with the information it needs. The set of procedures are as follows:

- UNIT IO informs synthesis of the number of unit input and output streams
- UNIT CLASS: Units are broken down into different classes which inform synthesis what kind of operation the unit performs on its input streams. The available classes are pressure change, temperature change, composition change, flow change.
- UNIT SVARS is the set of unit variables which are fixed for all designs, eg. ratio of operating to minimum reflux for a distillation column.

- UNIT DVARs is the set of unit design variables.
- UNIT CVARs is the set of unit calculated variables.
- UNIT SETDVAR sets the value of the unit design variables.
- UNIT DESIGN: Knowing the feed set and unit design variable values solves for the unit calculated variables and output streams.

As can be seen, information about the model is kept separate from the model design. The intention is that the unit design procedure should be able to access solvers or equation sets external to the synthesis environment if desired.

A.2.3 Incorporating unit models into synthesis

The information about units for a particular synthesis problem must be assembled. This is handled in the active units module (see section A.2.1) where each possible unit type is declared as an instance of a process unit. For the current problem, the user supplies the available unit types and the maximum number of instances of each. For a given unit type the model information is accessed. UNIT SVARs, UNIT CVARs and UNIT DVARs create space for and fill in all unit variable data. The user supplies the specific design variable information, i.e. the bounds, the number of discretisation levels and the type of discretisation (logarithmic or uniform). All information required for the design of a unit during synthesis now resides in the active units object. The algorithm for the design of a unit during synthesis is given in figure A.2.

A.3 Search objects

There are three search objects: problem, operator and solution. Since their definitions are interdependent, they are declared together in the same module - the synthesis problem module. This illustrates the limitation of using FORTRAN 90 along object-oriented lines as modules cannot be used to separately store the definitions of interlinked objects. The FORTRAN 90 structure for each search object (including commenting) is given.

The problem state has been described in some detail already. This is its final version incorporating all the information required for synthesis.

`! Full synthesis problem definition`

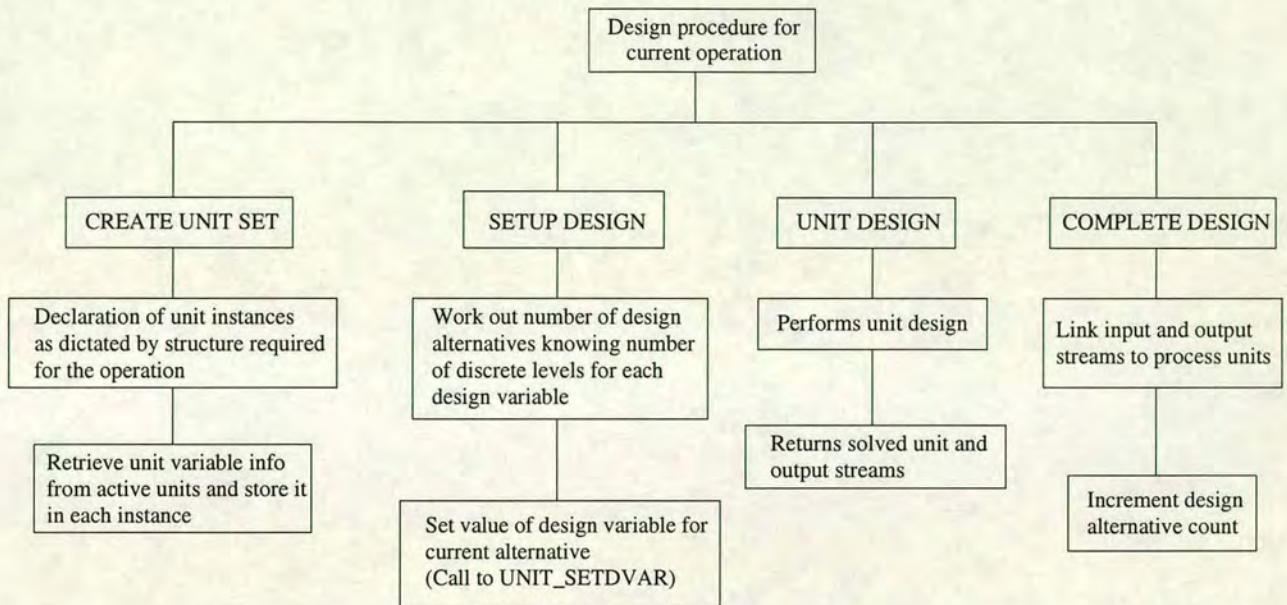


Figure A.2: Algorithm for performing unit design

```

type problem
! --- Basic problem definition ---
! PSF - basic problem definition/state, the process stream front
type(procstr_front), pointer :: psf
! ---- Targets to be satisfied ----
type(target_front), pointer :: tf
! --- Structure generation state ---
! Structure generation info/state
type(structgen_state), pointer :: sg_state
! --- Alternative enumeration state ---
! Alternative enumeration info - used to decide if operators are available
! for a given input
type(alternative_state), pointer :: alt_state
! --- Flowsheet state ---
! Flowsheet state - last stream and unit ids in the generated
! flowsheet
integer :: sid, uid
! --- Problem cost ---
! Cumulative cost to get to this problem state (from the root state)
real(kind=dp) :: cumul_cost
! --- Solutions to the problem ---
type(solution_list), pointer :: solution_list
end type problem

```

The operator identifies the current structure generation operation and current alternative type within that operation. It stores the designed process units and their total annualised cost.


```

type operator
  ! Type of structure generation operation
  integer :: otype
  ! Type of alternative being enumerated
  integer :: ptype
  ! Unit set created (if any)
  type(syunit_set), pointer :: puset
  ! Cost of the operation (= annualised unit set cost)
  real(kind=dp) :: cost
end type operator

```

A solution to a problem consists of the chain of operators and subproblems which go from the current problem to the solution state. Each problem may have more than one such solution.

```

type solution_list
  ! Solution information
  ! Value of the solution
  real(kind=dp) :: value
  ! Operator which transforms current subproblem into the next subproblem
  type(operator), pointer :: operator
  ! Next subproblem
  type(problem), pointer :: next_subprob
  ! The id of the solution to the next subproblem which this solution
  ! requires
  integer :: nsp_soln_id

  ! List section
  ! Solution ids assigned incrementally from 1.
  ! To set the current id we need to know the previous id.
  ! Store this and the size in the list header.
  integer :: last_id
  integer :: id
  integer :: rank
  integer :: size
  type(solution_list), pointer :: next
end type solution_list

```

A.4 Structure generation operations

The structure generation operations combined with the structure generation algorithm (stored in the structure generation state module) dictate the appearance and behaviour of the operators in the search tree. Each operation is stored in a FORTRAN 90 module as a set of procedures as follows:

- OPERATION DSS constructs the input set to the operation from the current process stream front.
- OPERATION FEASIBLE tests the input set for feasibility.
- OPERATION NEXTALTTYPE: Given the current alternative type, this works out the next alternative type.
- OPERATION OUTPUT: An operation may contain one or more such procedures, eg. the CT operation has CT SELECTION and CT DESIGN. These are the procedures which work out the output for the current alternative.

A generic structure generation module uses all the operation modules and is used to direct calls to the appropriate operations. Figure A.3 illustrates.

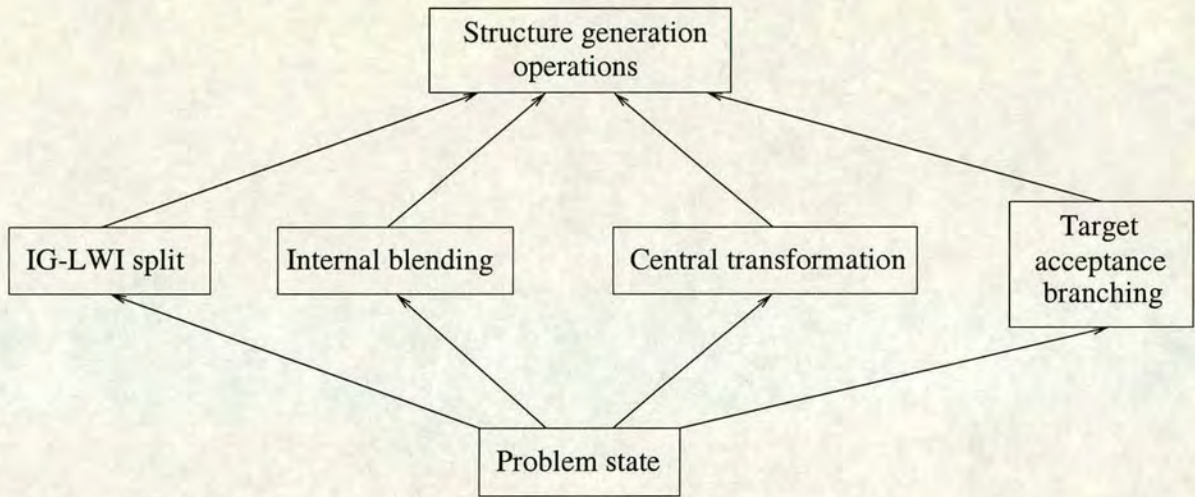


Figure A.3: Organisation of structure generation operations

A.5 Operator and solution handling

The modules which contain the operator and solution handling procedures sit above the structure generation operations as shown in figure A.4. The main procedures in the operator handling module are as follows:

- OPERATOR AVAILABLE checks if there is an operator available for the given input problem by looking at the alternative enumeration state of the input.
- OPERATOR NATURE establishes the structure generation operation type and alternative type by checking the input structure generation state and alternative enumeration state.

- OPERATOR INPUT constructs the input set for the current structure generation operation and tests its feasibility.
- OPERATOR OUTPUT makes the call to the appropriate structure generation output routine to get the output from the operator.
- OPERATOR CREATE is the subroutine which is responsible for creating the operators which appear in the synthesis tree. It first calls OPERATOR NATURE, then OPERATOR INPUT and finally OPERATOR OUTPUT.

The main procedures in the solution handling module are:

- SOLUTION STATE is a logical function which tests to see if the given subproblem is a solution, i.e. it checks if the process stream front is empty and that all targets are satisfied.
- SOLUTION EXISTS is a logical function which is set to true if the input problem has a solution associated with it. Two cases may set the function to true: firstly the input problem may have a set of solutions attached to it. Second, the input problem itself may be a solution. This is tested using SOLUTION STATE.
- SOLUTION VALUE works out the value of a solution, i.e. the product set value less the cumulative cost of the operators required to obtain the solution.
- SOLUTION LIST UPDATE organises the addition, removal and ranking of solutions to a given problem.

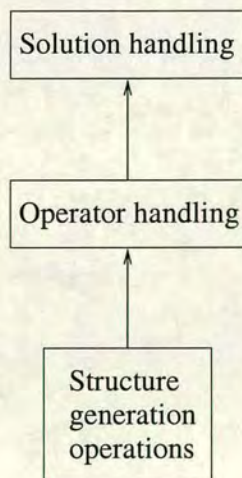


Figure A.4: Organisation of operator and solution handling

A.6 The search algorithm

The code for used by the tree search algorithm (with commenting) is shown below. The organisation of the search objects, the structure generation operations and the operator and solution handling procedures is such that it only has to use calls to the operator and solution handling to perform the search.

```

recursive subroutine tree_search(input)

  use solution_operations
  implicit none
  ! Arguments
  type(problem), pointer :: input
  ! Local variables
  type(problem), pointer :: output
  type(operator), pointer :: opn
  logical :: SUCCESS

  ! To create a tree path you first need an operator for the input problem
  do while (operator_available(input))

    call operator_create(input,opn,output,SUCCESS)

    if (SUCCESS) then
      ! The operation was successful and we have an output
      ! Test for a solution and test for active pruning.
      if (.not. solution_state(output) .and. &
          .not. prune_active(output))then
        call tree_search(output)
      else
        if (solution_state(output)) then
          ! Increment solution count
          soln_count = soln_count + 1
          ! Update the global solution set costs
          call GSC_update(output)
        elseif (prune_active(output)) then
          ! Deallocate opn and output
          call problem_deall(output)
          call operator_deall(opn)
        endif
      endif
    endif

    ! At the bottom of the path and popping up...
    ! Check for new solutions to the input problem - look for output
    ! problem solutions (which the input problem will not have seen yet)

```



```
    if (associated(output)) then
      ! We have an output to look at
      if (solution_exists(output)) then
        ! Update the solution list
        call solution_list_update(input,opn,output)
      else
        ! No new solutions encountered
        ! Clean out unwanted info
        call operator_deall(opn)
        call problem_deall(output)
      endif
    endif
  end do

end subroutine tree_search
```


Appendix B

Targetting Nonideal Separations

B.1 Introduction

This appendix details the integration of the nonideal separation targetting information into the synthesis method. The organisation of distillation line maps is first described followed by the changes made to the synthesis method implementation in order to enumerate the targetted separation alternatives.

B.2 Modelling distillation line maps

The hierarchy of information used to model distillation line maps is shown in figure B.1 and a description of each of the elements which comprise it is provided. This description includes the information which must be provided by the user and the procedures which are used during synthesis.

Component

The description of chemical component is comprised of its name, formula and a set of properties such as molecular weight, normal (1 atmosphere) boiling point, heat of formation etc. required for thermodynamics calculations. A table of chemical components is also defined and an instance of it declared to store the component table for a synthesis problem.

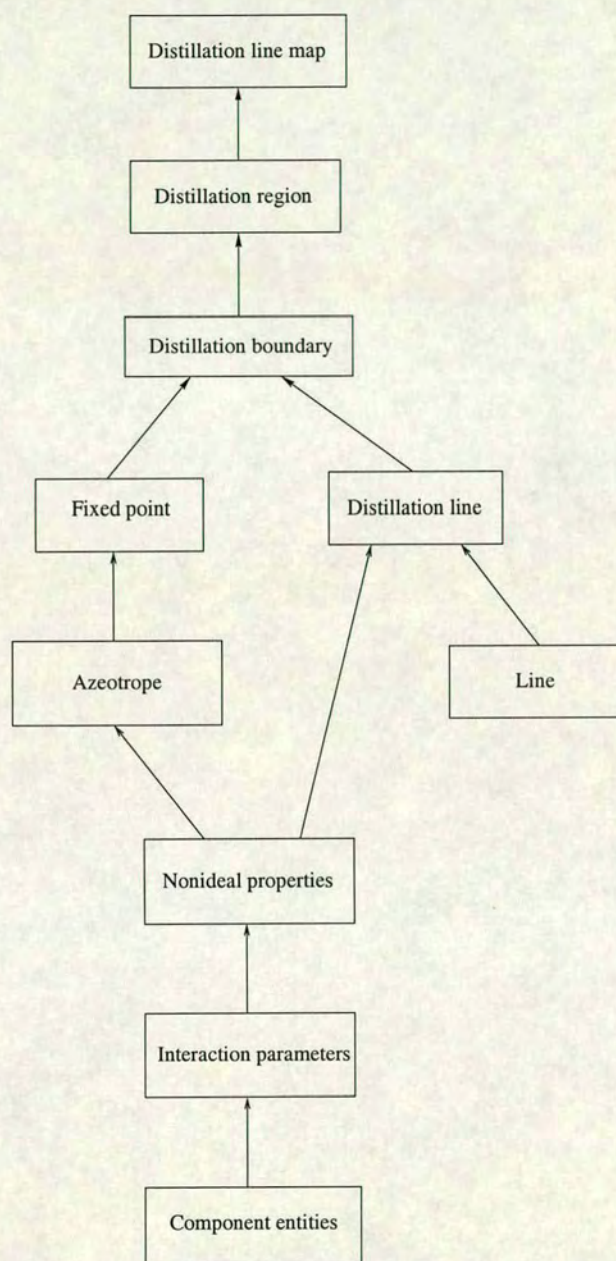


Figure B.1: Nonideal system information hierarchy

Component-component interactions

This is a matrix of the component-component interaction parameters required for the evaluation of the nonideal thermodynamic properties. An instance of the interaction matrix is declared to store the parameters for a particular synthesis problem.

Nonideal properties

This is a set of procedures used to perform nonideal vapour-liquid equilibrium calculations. The Wilson equation is used to model the liquid phase nonidealities.

Azeotrope

This describes an azeotrope by the number of constituent components, a name and its normal boiling point. An estimate for its composition is in the form of an indexed array to the problem component table. A table of azeotropes is defined and an instance of it declared to store the azeotrope table for a synthesis problem. Procedures are provided to:

- Prompt for and read in the data for the problem azeotrope table provided by the user - the number of azeotropes and the information required to describe each of them.
- Retrieve information from this structure.
- Determine an accurate value for the azeotropic composition at different pressures. This procedure works for binary azeotropes only.

Fixed point

This describes a distillation fixed point as either a component or an azeotrope with an index to the appropriate species table. The species normal boiling point is included as it is useful for ordering purposes. The user supplies the type of each fixed point, i.e. stable, unstable or saddle point. The fixed points for a given problem are stored as a list, ranked in order of increasing normal boiling point. Procedure are provided for :

- Data input (by the user), retrieval and output.
- Testing if a given composition corresponds to a fixed point at a given pressure.

Line

The introduction of a data type for a line and associated procedures was found to be necessary for dealing with distillation path and mass balance intersections. A line is described as set of linear equations. Each equation consists of a coefficient matrix and a constant. The equations can either be in parametric form or normal form. The former

is preferred for this work as it stores the start and end points of the line. Procedures are provided to:

- Construct line equations from two points, in either normal or parametric form.
- Determine the intersection of two lines.
- Test if a given point is on a given line.
- For parametric lines test if given point is both on the line and within the start and end points of the line.

Distillation line

A distillation line is described by a list of composition nodes determined by a total reflux calculation. Each node has a composition and a link to the next and previous nodes. The list is accessed via the most volatile and least volatile composition nodes. The straight line approximation to the path, i.e. the line from the most and least volatile composition nodes, is also included. Procedures are provided to:

- Construct a distillation line from a given composition at a given pressure. This is achieved by performing bubble point calculations starting from the input composition until no more change in composition occurs. This is repeated using dew point calculations. The full path is the union of these two partial paths.
- Determine if two paths have two same start and end points. This simply involves checking if their terminal compositions are the same (within a tolerance).

Distillation boundary

A distillation boundary is a special case of a distillation line. In addition to the path information, the fixed points which characterize the boundary are provided by the user and stored as pointers to the fixed point list. Procedures are provided to:

- Read in boundary information from the user.
- Determine the boundary path for a given pressure. This is limited to boundaries involving binary azeotropes and pure components. Starting from a specified side of a binary azeotrope and adding a small amount of the nonazeotropic component to create a ternary mixture, bubble or dew point calculations (depending on whether the pure component end of the boundary is lighter or heavier than the azeotrope end) are carried out until the pure component boundary fixed point is reached. The path obtained is very close approximation to the distillation

boundary path. It is completed by replacing the ternary starting point with the binary azeotropic composition.

Distillation region

A distillation region consists of a list of fixed points (ranked in order of increasing normal boiling point) and a list of distillation boundaries. The former point to the fixed point list for the system while the latter is created for the given region. The regions which describe a given system are stored as a list. Procedures are provided to:

- Read in region information provided by the user, i.e. the set of fixed points and the distillation boundary which make up the region.
- Retrieve the most volatile and least volatile fixed points in a given region.
- Retrieve the compositions of the most volatile and least volatile fixed points.
- Determine the region with which the given region shares a distillation boundary.

Distillation line map

The triangular diagram used to represent ternary nonideal systems is the surface defined by the mole fraction summation equation and by the composition limits, i.e. all mole fractions are between 0.0 and 1.0. The summation equation removes one dimension, allowing a three component problem to be represented by a two dimensional diagram. For this work the dimensionality is reduced by declaring the composition of one of the components redundant. It is omitted from all line equations and its value is obtained by solving the summation equation. The bounds of the diagram are described by a set of lines.

Using the maps

For this work it is assumed that the user possesses all the information on the problem system. He/she supplies the fixed points and defines the distillation regions and their boundaries. The data for a particular system is assembled before synthesis, with the exception of the azeotrope composition and distillation boundary paths which are capable of being determined at different pressures. The end result is a distillation line map which can be both referred to during synthesis. It is structured, problem-specific information which can be used to target the separation unit design alternatives.

B.3 Integrating targetted alternatives into the synthesis problem state

Recall that the synthesis problem state consists of four components: the target front, process stream front, structure generation state and alternative enumeration state (section A.2). The targetting implemented for this work affects the last of these. It takes over the assignment of unit design variable values from discretisation (section 3.4). Therefore, changes must be made in order to both enumerate and assign the targetted design alternatives during synthesis.

The unit design variable set is now composed of two subsets: the variables set by discretisation or discrete variables and those set by targetting, the targetted variables. The integration of targetting into the design alternative state almost cover identification, enumeration and assignment.

- New information is added to the unit variable structure. Design variables for a unit are given a flag indicating whether they are to be set by discretisation or a by targetting procedure.
- When enumerating the design alternatives, the targetted block of variables is considered to be equivalent to a single discrete variable, i.e. it possesses a number of design levels and each must be explored in turn. The first variable or head of the targetted block stores the number of design alternatives and the current design level. The former value is supplied by a routine within the targetting procedure. An *effective* number of discrete variables is determined, equal to the number of actual discrete variables plus the number of targetted alternatives. This permits the design alternative state to be constructed and enumerated in the same manner as before.
- Targetting assigns values to a group of variables simultaneously. The assignment occurs after that of the discrete variables in case the targetting procedure requires the discrete variable values as inputs. To facilitate this, the targetted group or block of variables is placed after the discrete block of variables in the unit model design variable set. At the moment this must be done by the user writing that model. Within the unit model SETDVAR procedure (section A.2.2) the user again puts in place the appropriate calls to assign the variables by discretisation or by targetting.

The targetting procedure thus performs two functions. Firstly, it works out the number of targetted design alternatives. Secondly, for each alternative, it sets the values of the variables within the targetted block. These functions combined with the treatment of the targetted block of variables as a single discrete variable allow targetting to integrate easily with the existing design variable assignment algorithm (section A.2.3). The position of the targetting procedure within the synthesis hierarchy is shown in figure B.2.

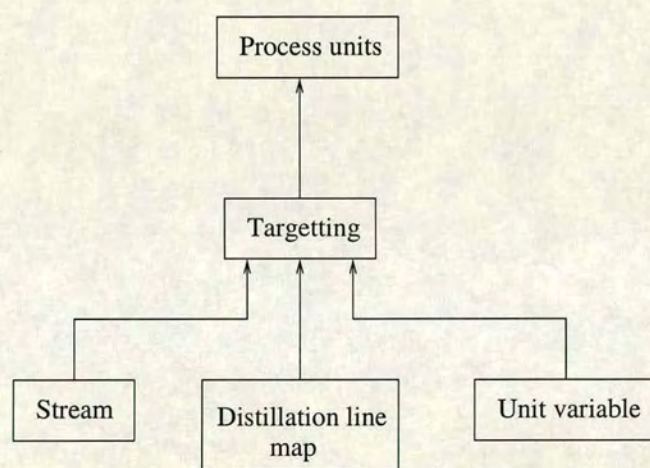


Figure B.2: Position of targetting in the synthesis hierarchy

Appendix C

Nonideal Separation Column Model

C.1 Introduction

The model used for the design of nonideal separations mimics the Fenske-Underwood-Gilliland design method for ideal separations but employs rigorous activity-based thermodynamics to model the vapour-liquid equilibrium. Since the relative volatility of components is not constant, the use of the Fenske minimum number of stages calculation for nonideal systems requires that the relative volatility must be calculated for each stage. This has resulted in a stage-by-stage infinite reflux calculation which employs an iterative solution method to satisfy both the separation requirements and the column mass balance. This appendix details the algorithm used to determine the minimum number of stages.

C.2 Determination of the minimum number of stages

Figure C.1 shows the algorithm used to perform the infinite reflux calculation. It is intended for ternary systems and, given the recoveries of the two key components, it iterates on the unknown nonkey recovery until both the mass balance and separation requirements are satisfied. For any value of the nonkey recovery, a stage-by-stage infinite reflux calculation is carried out until the required keys separation is obtained. The output streams are then tested to see if they satisfy the mass balance. Convergence of the mass balance is carried out by first determining the bounds on the nonkey recovery needed for its satisfaction. Once these bounds have been found, the bisection method solves the final value of the nonkey recovery.

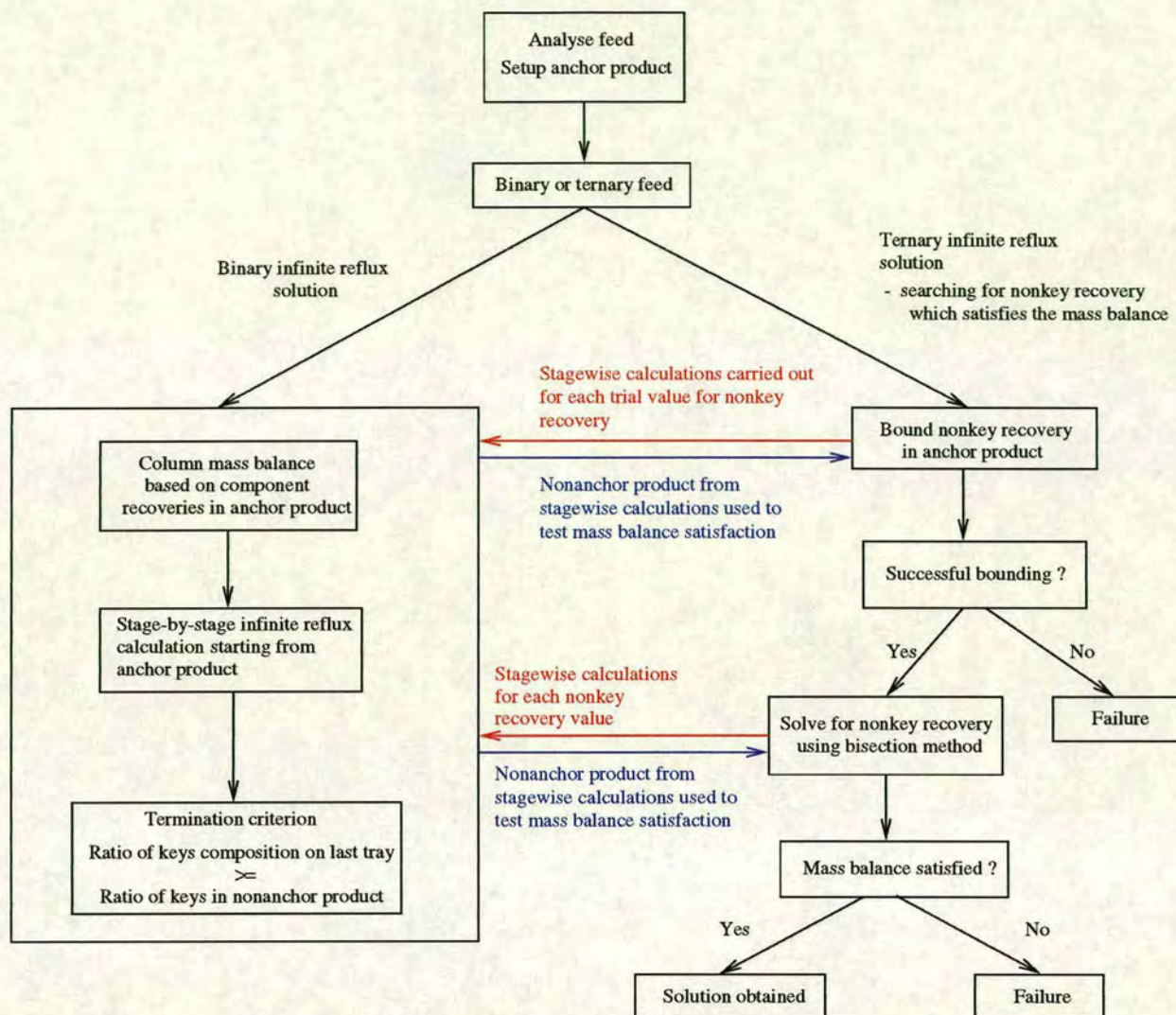


Figure C.1: Infinite reflux algorithm

C.2.1 Setting up

The column feed is analysed to determine the number of components present. If there are less than two then no separation is necessary. Two component feeds invoke the binary solution method and three components the ternary solution method. Both cases determine the minimum number of stages through the stagewise infinite reflux calculation. However, the latter also solves for the nonkey recovery. The key component recoveries are set for the specified anchor product. For ternary mixtures, the nonkey recovery is initialised to 0.0 or 1.0 depending on whether the component is a light nonkey or a heavy nonkey.

C.2.2 Minimum number of stages algorithm

This is the subroutine which carries out the infinite reflux calculation to determine the minimum number of stages required by the separation between the key components.

1. Given the feed stream and anchor product component recoveries, determine the distillate and bottoms composition. The concept of anchor products is described in section 6.3.1.
2. The termination criterion for the subroutine is the keys separation in the nonanchor product. For the case where the anchor product is the bottom product and the number of stages is determined by bubblepoint calculations, the termination criterion is given by:

$$y_{lk,d}/y_{hk,d} \quad (C.1)$$

where $y_{lk,d}$ is the composition of the light key component in the distillate and $y_{hk,d}$ is the composition of the heavy key component in the distillate. For the case where the anchor product is the distillate, the termination criterion is given by:

$$x_{hk,b}/x_{lk,b} \quad (C.2)$$

where $x_{hk,b}$ is the composition of the heavy key component in the bottoms and $x_{lk,b}$ is the composition of the light key component in the bottoms.

3. Set the composition on starting stage to be the anchor product composition.
4. Depending on the anchor product carry out a bubble or dew point calculation to obtain the stage temperature and equilibrium composition.
5. Make the input composition to the next stage equal to the equilibrium composition calculated for the current stage.
6. Test the termination criterion for the current stage. If the keys separation in the output from the last stage added is greater than that given by the termination criterion, stop the calculation. The number of stages required has been found.
7. Otherwise go back to (4) and repeat.

C.2.3 Mass balance algorithm

The assumption of sharp and semi-sharp separations means the nonkey component is not expected to distribute. If the mass balance is close to a solution for the initial value of the nonkey recovery in the anchor product (either 0.0 or 1.0) then a solution is declared. However, if the mass balance error is significant then iteration on the nonkey recovery takes place.

For a given value of the nonkey recovery the number of stages algorithm calculates the mass balance error, MASSBAL. For the trial value of the nonkey recovery a mass balance is carried out which yields a value for the nonkey mole fraction in the nonanchor product. MASSBAL is the difference between the latter nonkey composition and the corresponding value obtained from the stage-by-stage calculation. The algorithm employed to achieve convergence of the mass balance is:

1. If the starting value of the nonkey recovery is 1.0 then its solution value is less than or equal to this value. The reverse applies if the starting value is 0.0.
2. Bound the solution by taking successive small steps in the appropriate direction. For each value of the nonkey recovery the minimum number of stages procedure is performed to obtain a value for MASSBAL.
3. The solution is bounded when the sign for MASSBAL reverses. If the solution cannot be bounded (either the recovery goes out of bounds or stage calculation fails) then a solution cannot be found, otherwise...
4. The bisection method is used to determine the value for the nonkey recovery.

Appendix D

Acroynm List

Introduction

MCP multicomponent product

MCP Literature Review

CAD	component assignment diagram
DSB	directional separation breakpoint
MAD	material allocation diagram
mMAD	modified material allocation diagram
MCP	multicomponent product
MILP	mixed integer linear program
MINLP	mixed integer nonlinear program
NLP	nonlinear program
R-matrix	component recovery matrix
S-matrix	separation task matrix
SAD	stream allocation diagram
SPM	separation product matrix

MCP Synthesis

CT	central transformation
IG	immediate gratification stream
LWI	loiter-with-intent stream
IB	internal blending
MCP	multicomponent product
NS	new stream
PSF	process stream front
TAB	target acceptance branching

Nonideal Synthesis

RS-NS	recycle newstream
RS-IG	recycle immediate gratification stream
RS-LWI	recycle loiter-with-intent stream
RC	recycle creation
RS	recycle split
RB	recycle blending

References

- [1] Zainuddin A.-M. *Process Synthesis for Waste Minimisation With Emphasis on the Synthesis of Cleaner and Cost Effective Distillation Sequences for Azeotropic Mixtures*. PhD thesis, University of Edinburgh, 1998.
- [2] A. Aggarwal and C.A. Floudas. Synthesis of general distillation sequences - nonsharp separations. *Computers and Chemical Engineering*, 14(6):631–653, 1990.
- [3] M.J. Andreovich and A.W. Westerberg. An milp formulation for heat integrated distillation sequence synthesis. *AIChEJ*, 31:1461, 1985.
- [4] G. Bamopoulos, R. Nath, and R.L. Motard. Heuristic synthesis of nonsharp separation sequences. *AIChEJ*, 34(5):763–780, 1988.
- [5] M.H. Bauer and J. Stichlmair. Superstructures for the mixed integer optimisation of nonideal and azeotropic distillation processes. *Computers and Chemical Engineering*, 20(Suppl.):S25–S30, 1996.
- [6] Y.-M. Chen and L.T. Fan. Synthesis of complex separation schemes with stream splitting. *Chem. Eng. Sci.*, 48(7):1251–1264, 1993.
- [7] S.-H. Cheng and Y.A. Liu. Studies in chemical process design and synthesis: 8. A simple heuristic method for the synthesis of initial sequences for sloppy multicomponent separations. *Ind. Eng. Chem. Res.*, 27(12):2304–2322, 1988.
- [8] M.F. Doherty and G.A. Caldarola. Design and synthesis of homogeneous azeotropic distillations. 3: The sequencing of columns for azeotropic and extractive distillations. *Ind. Eng. Chem. Fundam.*, 24:474–485, 1985.
- [9] M.F. Doherty and J.D. Perkins. On the dynamics of distillation processes - 1, The simple distillation of multicomponent non-reacting, homogeneous liquid mixtures. *Chemical Engineering Science*, 33:281–301, 1978.
- [10] Z.T. Fidkowski, M.F. Malone, and M.F. Doherty. Computing azeotropes in multicomponent mixtures. *Comp. Chem. Eng.*, 17(12), 1993.
- [11] C.A. Floudas. Separation synthesis of multicomponent feed streams into multicomponent product streams. *AIChEJ*, 33(4):540–550, 1987.

- [12] E.R. Foucher, M.F. Doherty, and M.F. Malone. Automatic screening of entrainers for homogeneous azeotropic distillation. *Ind. Eng. Chem. Res.*, 30:760–722, 1991.
- [13] E.S. Fraga and K.I.M. McKinnon. CHiPS: A process synthesis package. *Chemical Engineering Research and Design*, 72(A3):389–394, 1994.
- [14] F. Friedler, L.T. Fan, and B. Imreh. Process network synthesis: Problem definition. *Networks*, 31(2):119–124, 1998.
- [15] F. Friedler, K. Tarjan, Y.W. Huang, and L.T. Fan. Graph-theoretic approach to process synthesis - polynomial algorithm for maximal structure generation. *Computers and Chemical Engineering*, 17(9):929–942, 1993.
- [16] A. Herron. Azeotropic modelling in process synthesis. Master's thesis, Department of Chemical Engineering, The University of Edinburgh, 1998.
- [17] E.J. Hoffmann. *Azeotropic and extractive distillation*. Interscience Publishers, John Wiley and Sons, 1964.
- [18] C.A. Jaksland, R. Gani, and K.M. Lien. Separation process design and synthesis based on thermodynamic insights. *Chemical Engineering Science*, 50(3):511–530, 1995.
- [19] W. Kistiakowsky. *Zeitschrift fur Physikalische Chemie*, 107:65, 1923.
- [20] L.N. Laroche, N. Bekiaris, H.W. Andersen, and M. Morari. The curious behaviour of homogeneous azeotropic distillation - implications for entrainer selection. *AIChEJ*, 38(9):1309–1328, 1992.
- [21] Z.-Y. Liu. Matrix-based heuristic synthesis of distillation sequences with several feed streams and multicomponent products. *Chemical Engineering Research and Design*, 76(A3):302–307, 1998.
- [22] Z.-Y. Liu and X.-I. Xu. Heuristic procedure for the synthesis of distillation sequences with multicomponent products. *Chemical Engineering Science*, 50(12):1997–2004, 1995.
- [23] Z.-Y. Liu and X.-I. Xu. Matrix-based heuristic synthesis for production of mixed products via sharp distillation sequences with bypass. *Chemical Engineering Research and Design*, 73(A):13–19, 1995.
- [24] M.F. Malone and M.F. Doherty. Separation system synthesis for nonideal liquid mixtures. In Doherty M.F. Biegler L.T., editor, *FOCAPD 1994*, volume 91 of *AIChE Symposium Series*, pages 9–18, 1994.
- [25] Economic indicators. *Chemical Engineering*, October 1974.
- [26] Economic indicators. *Chemical Engineering*, May 1995.
- [27] M. Muraki and T. Hayakawa. Separation process synthesis for multicomponent products. *Journal of Chemical Engineering of Japan*, 17(5):533–538, 1984.

- [28] M. Muraki and T. Hayakawa. Synthesis of a multicomponent multiproduct separation process with nonsharp separators. *Chem. Eng. Sci.*, 43(2):259–268, 1988.
- [29] M. Muraki, K. Kataoka, and T. Hayakawa. Evolutionary synthesis of a multicomponent multiproduct separation process. *Chem. Eng. Sci.*, 41(7):1843–1851, 1986.
- [30] N. Nishida, G. Stephanopoulos, and A.W. Westerberg. A review of process synthesis. *AIChEJ*, 27(3):321–351, 1981.
- [31] R.H. Perry and D. Green, editors. *Perry's Chemical Engineers' Handbook*. McGraw-Hill, 1984.
- [32] R. Raman and I.E. Grossman. Integration of logic and heuristic knowledge in MINLP optimization and process synthesis. *Computers and Chemical Engineering*, 16(3):155–171, 1992.
- [33] R.N.S. Rathore, K.A. van Wormer, and G.J. Powers. Synthesis strategies for multicomponent separation systems with energy integration. *AIChEJ*, 20(3):491–502, 1974.
- [34] T. Senos Matias. *Approximate Procedures for Simulation and Synthesis of Nonideal Separation Systems*. PhD thesis, The University of Edinburgh, 1996.
- [35] C.M. Sheppard. *Constraint-Directed Non-Sharp Separation Sequence Design*. PhD thesis, Washington University, 1989.
- [36] E.M.B. Smith and C.C. Pantelides. Design of reaction/separation networks using detailed models. *Computers and Chemical Engineering*, 19(Suppl.):S83–S88, 1995.
- [37] E.M.B. Smith and C.C. Pantelides. Global optimisation of nonconvex MINLPs. *Computers and Chemical Engineering*, 21(Suppl.):S791–S796, 1997.
- [38] J.M. Smith and H.C. Van Ness. *Introduction to Chemical Engineering Thermodynamics*. McGraw-Hill, 1987.
- [39] J.G. Stichlmair. Separation of ternary mixtures by rectification. *Int. Chem. Eng.*, 31:423–433, 1991.
- [40] J.G. Stichlmair, J.R. Fair, and J.L. Bravo. Separation of azeotropic mixtures via enhanced distillation. *Chemical Engineering Progress*, 63(1989), 85.
- [41] J.G. Stichlmair and J.R. Herguijuela. Processes of zeotropic and azeotropic distillation. *AIChEJ*, 38:1523–1535, 1992.
- [42] O.M. Wahnschafft, T.P. Jurain, and A.W. Westerberg. SPLIT: A separation process designer. *Computers and Chemical Engineering*, 15(8):565–581, 1991.

- [43] O.M. Wahnschafft, J.W. Kohler, and A.W. Westerberg. Homogeneous azeotropic distillation: Analysis of separation feasibility and consequences for entrainer selection and column design. *Comp. Chem. Eng.*, 18(suppl.):S31–S35, 1994.
- [44] O.M. Wahnschafft, J.-P. Le Rudulier, and A. W. Westerberg. A problem decomposition approach for the synthesis of complex separation processes with recycles. *Ind Eng Chem Res*, 32:1121–1141, 1993.
- [45] R.R. Wehe and A.W. Westerberg. An algorithmic procedure for the synthesis of distillation sequences with bypass. *Computers and Chemical Engineering*, 11(6):619–627, 1987.